(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 February 2001 (15.02.2001)

PCT

(10) International Publication Number
WO 01/11503 A2

(51) International Patent Classification⁷: G06F 17/30

(21) International Application Number: PCT/US00/21702

(22) International Filing Date: 8 August 2000 (08.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

09/372,402	10 August 1999 (10.08.1999)	US
09/371,161	10 August 1999 (10.08.1999)	US
09/372,410	10 August 1999 (10.08.1999)	US

(71) Applicant: AKAMAI TECHNOLOGIES, INC. [US/US]; 500 Technology Square, Cambridge, MA 02139 (US).

(72) Inventor: TSIMELZON, Mark; Apt. 313, 745 South Bernardo Avenue, Sunnyvale, CA 94087 (US).

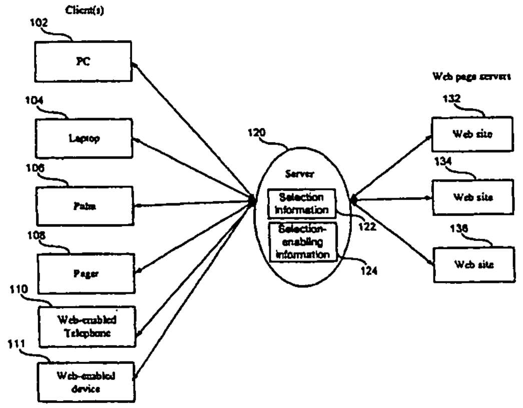
(74) Agents: MAJERUS, Laura; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 et al. (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

(54) Title: METHOD AND APPARATUS FOR NOTIFYING A USER OF CHANGES TO CERTAIN PARTS OF WEB PAGES



(57) Abstract: A method and apparatus that allows users to select certain sub-elements of web pages and to turn notifications on or off for those selected sub-elements of the web page. A notification is sent when the selected sub-element of the web page changes in a user-specified way. The user sets a notification condition for each selected sub-element of the web page. When the notification condition is true for any of the selected sub-elements, the system notifies the user that his selected sub-element of the web page has changed in the user-selected manner. Thus, certain changes to the web page may not cause a notification to occur. A described embodiment of the present invention breaks the web page into sub-elements of varying granularity. The notification criteria includes a frequency at which the page should be monitored for changes. The user is notified that the notification condition has been met for a sub-element via a user-selected notification mechanism.

WO 01/11503 A2

Method and Apparatus for Notifying a User of Changes to Certain Parts of Web Pages

RELATED APPLICATIONS

BACKGROUND OF THE INVENTION

The present invention relates generally to computer networks and, specifically, to a method and apparatus that allow users to select certain portions (sub-elements) of web pages to form "short" web pages.

The expanded popularity of the World Wide Web ("the web") has brought new problems for web users. As users obtain more and more information from the web, they must visit greater numbers of different web pages to obtain the information. This information is often contained on several separate web pages or web sites. Alternately, a user may wish to see only a small piece of information that is contained in a very large and complex page or site. The user must search through the pages in which he is not interested to reach the information that he wishes to see.

Many users visit the same sequence of web pages or sites on a daily or a regular basis. For example, some users might check the weather or their stock portfolio on a daily basis. Even though a user may visit the same web pages and/or sites regularly, the user must still take individual actions to visit each of his regularly visited web pages or sites. For example, the user may have to enter the web addresses of each page or click on each page in a list of saved addresses in his browser. This repeated entry of web addresses is time consuming and involves un-needed repetitive action by the user.

What is needed is a way to avoid regularly re-entering the same multiplicity of web addresses and a way to avoid having to navigate through multi-level web sites to reach desired information or to learn when the desired information has changed.

SUMMARY OF THE INVENTION

A described embodiment of the present invention allows users to select certain portions (sub-elements) of web pages and to turn notifications on or off for those selected sub-elements of the web page. A notification is sent when the selected sub-element of the web page changes in a user-specified way. The user sets a notification condition for each selected sub-element of the web page. When the notification condition is true for any of the selected sub-elements, the system notifies the user that his selected sub-element of the web page has changed. Thus, certain changes to the web page may not cause a notification to occur. For example, the change may not occur in a selected element of the web page. As another example, a selected element of the web page may have changed, but the change may not meet the notification condition set by the user. The user can use any of a wide variety of client devices to view the web page, such as a computer, a handheld device, a cell phone, an alphanumeric pager, or any appropriate web-enabled device or appliance.

The described embodiment of the present invention breaks the web page into sub-elements of varying granularity. Selection-enabling information is added to the web page to enable the user to select the sub-elements and the web page is sent to the user's browser. The user selects certain sub-elements or sub-elements of the web page and sets notification criteria for each. The user's choices are stored, preferably in the server, although they could also be stored at another location, such as the user's system. The notification criteria includes a frequency to monitor for changes. The server preferably retrieves the web page in accordance with the notification frequency or frequencies set by the user. The server then checks whether the notification condition has been set for each sub-element having a notification condition.

The user is notified that the notification condition has been met for a sub-element via a user-selected notification mechanism. Notification mechanisms include, but are not limited to e-mail messages, paging the user, and placing a notification on a notification web page. The user preferably selects one of several possible notification methods.

In accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method to notify a user of certain changes to a web page, comprising: allowing a user to choose sub-elements on a web page as subjects of notification; saving the user's choices; and monitoring the user's chosen sub-elements on the web page and notifying the user when a notification condition is true for at least one of the sub-elements.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to an apparatus that notifies a user of certain changes to a web page, comprising: a software portion configured to allow a user to choose sub-elements on the web page as subjects of notification; a software portion configured to save the user's choices; and a software portion configured to monitor the user's chosen sub-elements on the web page and to notify the user when a notification condition is true for at least one of the sub-elements.

Advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram of data processing elements in accordance with a preferred embodiment of the present invention.

Figs. 2(a) and 2(b) are block diagrams showing an information flow in a preferred embodiment of the present invention.

Fig. 3(a) shows a display of a startpage that allows a user to create and edit shortpages and combopages.

Figs. 3(b)-3(e) show the HTML source for a startpage.

Figs. 4(a)-4(c) are flow charts showing how a client and a server interact to perform shortpage operations, such as create, edit, delete, and view.

Fig. 5(a) shows a display of an exemplary web page.

Figs. 5(b) and 5(c) show the web page of Fig. 5(a) displayed along with selection-enabling information.

Fig. 5(d) shows the web page of Fig. 5(b) with a different level of block detail.

Fig. 5(e) shows a web page that allows the user to view/edit shortpage properties.

Fig. 6 is a flow chart showing how the client and the server interact to allow the user to create/edit a shortpage.

Fig. 7 shows an example of HTML parsing used to create/edit shortpages.

Fig. 8 shows an example of a split page method used to create/edit shortpages.

Fig. 9 is a flow chart of how to determine a display level when creating shortpages.

Fig. 10 shows the startpage of Fig. 3(a) after a shortpage has been created based on the web page of Fig. 5(a).

Fig. 11 shows the shortpage created from the web page of Fig. 5(a).

Fig. 12 shows a fullpage corresponding to the web page of Fig. 5(a).

Fig. 13 is a flow chart showing how the client and server interact to allow the user to view a shortpage.

Fig. 14 is a flow chart showing details of determining whether a URL is the URL of a shortpage.

Fig. 15(a) shows a display for creating a combopage.

Fig. 15(b) shows an example of a combopage.

Fig. 16(a) is a flow chart for creating a combopage.

Fig. 16(b) is a flow chart for displaying a combopage.

Fig. 16(c) shows a web page that allows the user to view/edit combopage properties.

Fig. 17(a) shows a display that allows a user to edit notifications.

Fig. 17(b) shows a display that allows a user to edit notification properties.

Fig. 17(c) shows a notification page on which notifications of changes are displayed.

Figs. 18(a) and 18(b) are flow charts showing notification operation.

Fig. 19 is a flow chart for viewing a shortpage when the client is a personal digital assistant.

Fig. 20 is a block diagram of data flow when a shortpage or a combopage is viewed on a personal digital assistant.

Fig. 21 is a display of a shared portal shortpage.

DETAILED DESCRIPTION OF EMBODIMENTS

Reference will now be made in detail to several embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever practicable, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

I. General Discussion

A described embodiment of the present invention allows users to select certain sub-elements of one or more web pages as a shortpage. The user selects certain sub-elements of an original web page to create a shortpage. The user's selection information is saved and, when the user views the shortpage at a later time, only the sub-elements of the web page indicated by the user are displayed for viewing by the user. Thus, the user views only preselected sub-elements of the web page. "Combopages" (short for "combination pages") are created by combining shortpages or from combining selections from more than one web page or site. This section discusses an exemplary data processing system used to implement a preferred embodiment of the present invention.

Fig. 1 is a block diagram of data processing elements in accordance with a preferred embodiment of the present invention. Fig. 1 includes a server data processing system 120 communicating with one or more client data processing systems. These client data processing systems include, but are not limited to, a desktop personal computer (PC) 102, a laptop computer 104, a palm computer (personal computer/assistant or handheld device) 106, a pager 108 (such as an alphanumeric pager), a web-enabled telephone or a cell phone 110, or some other appropriate web-enabled device or appliance 111. A web-enabled telephone or device could use, for example, the WAP (Wireless Application Protocol) or any other appropriate protocol. It should be understood that the client data processing systems shown in Fig. 1 are shown for purposes of example only and are not to be construed in a limiting sense. Any other appropriate type of client can be used in conjunction with the present invention. Fig. 1 also includes a plurality of web page servers 132, 134, 136. Each web page server communicates with server 120 and stores one or more web sites or web pages. Server 120 stores selection information 122 for each user and further stores selection-enabling information 124 that is added to a web page to enable a user to make selections from a web page. Communication can be over a network, such as the internet, an intranet, a wireless network, or in any other appropriate manner.

Fig. 2(a) shows an overview of creation of a shortpage. As shown in Fig. 2(a), during operation, a client 202 sends a request to server 120 for a web page. Server 120 retrieves the web page from an appropriate one of web sites 132-136. The server 120 adds selection-enabling information to the retrieved web page and sends the web page to the client. The client selects sub-elements of the web page and sends that selection information to server 120. Server 120 stores selection information 122 describing the shortpages defined by a user.

Fig. 2(b) shows an overview of viewing a shortpage. The client sends a request for a shortpage (or a combopage) to server 120. Server 120 determines the page or pages needed to view the shortpage in accordance with the stored selection information and retrieves the page or pages from the web site 132. Server 120 shortens the retrieved web page in accordance with the selection information to form a shortpage and sends the shortpage to the client.

It should be understood that each of the clients, servers, and web servers in the described embodiment includes a processor and a memory. The memory includes instructions capable of being executed by the processor to perform the functions described below. A client and/or a server can also include a computer readable medium for storing the instructions. Server 120 communicates with the clients and the web page servers via any appropriate communication mechanism, including but not limited to a network, an intranet, the internet, wireless communications, telecommunications, cable modems, and satellite communications.

A client may include browser software suitable for viewing web pages. The browser software can be standalone or integrated within other software products. The functionality can be stored, for example, as a link, a JavaScript, or as a Java applet. Handheld clients contain a specialized browser that receive "snipped" sub-elements of web pages for viewing on a handheld client. Other clients (such as cell phones) do not necessarily contain a browser. It should be understood that references herein to "HTML" can be understood to refer to any appropriate page specification language, such as a hypertext language or a proprietary language.

The following paragraphs describe an example of how a user makes a shortpage. This example is provided for exemplary purposes only and is not to be interpreted in a limiting sense.

Fig. 3(a) shows a display of a "startpage" on a browser in a client. The startpage allows a user to create and edit shortpages and combopages and displays existing shortpages and combopages. Each user has his own startpage. The information needed to make a startpage is preferably stored in server 120. Fig. 3(a) shows a URL 310 of the exemplary startpage for user John Smith

(http://www.shortware.com:9999/UserName=Jsmith&ProcName=StartPage).

The startpage of Fig. 3(a) includes a link to a Notifications page 312, a link to an Edit profile page 314, and a link to a Help page 316. The startpage of Fig. 3(a) allows a user to create, edit, delete, and view shortpages and their options. To create a shortpage, in the described embodiment, the user enters a URL of a page into area 318 and selects "Create Shortpage" button 320. After a shortpage is created, its name will be displayed in Shortpages column 332. An existing shortpage can be viewed ("Go"), edited, options edited, or deleted by selecting the corresponding action in action box 330 and selecting a name of an existing shortpage. Creation, editing and deleting of notifications and combo pages are discussed in detail below.

Figs. 3(b)-3(e) are an example of the HTML source code for an exemplary startpage, similar to the StartPage of Fig. 3(a). This source code is included for the purpose of example and is not to be taken in a limiting sense.

Figs. 4(a)-4(c) are flow charts showing how a client and a server interact to perform shortpage operations, such as create, edit, delete, and view. Figs. 4(a) and 4(b) show a method performed by the client. In the described embodiment, the functionality of Figs. 4(a) and 4(b) are implemented via a Java Script executed by the client browser, although any appropriate implementation can be used. Fig. 4(c) shows a method performed by server 120.

In Figs. 4(a) and 4(b), the client receives the user's actions (e.g., entering the URL of a web page and clicking "Create shortpage") and determines whether the user wants to create a shortpage (element 410), edit a shortpage (element 420), view a shortpage (element 430), delete a shortpage (element 440), or edit an option (element 450).

If the user wants to create a new shortpage (element 410), the client sends the URL of the page to shorten and a request to create a shortpage to server 120. The client receives a page created from the requested page and from selection-enabling information. In the example, the selection-enabling information allows the user to indicate that sub-elements of the page are shown or hidden. The client then displays the page, and allows the user to create a shortpage as described below. The resulting selection information is sent to server 120. In the described embodiment, selection information is sent to the server each time a page element is marked as "shown" or "hidden." In return, a new preview of the shortpage is sent from the server to the client, so that the preview view of the short page reflects the currently shown/hidden element. The server adds the name of the shortpage to the list of shortpages 332 on the page.

If the user wants to edit a shortpage (element 420), the client sends the URL of the shortpage to edit and a request to edit a shortpage to server 120. The client receives a shortpage created from the original fullpage including selection-enabling information. The selection enabling information includes show/hide buttons and header links. The client then displays the page upon which the shortpage is based, including the selection-enabling information, and allows the user to edit the shortpage as described below. The resulting selection information is sent to server 120. Note that the show/hide buttons can instead be implemented as a one-button user interface, or by using some other appropriate type of user interface that allows the user to choose a subset of the web page elements.

If the user wants to view a shortpage (element 430), the client sends the URL of the shortpage to view and a request to view a shortpage to server 120. The client receives the shortpage from server 120. The shortpage to be viewed does not include any selection-enabling information. The client then displays the shortpage.

If the user wants to delete a shortpage (element 440), the client sends the URL of the shortpage to delete and a request to delete a shortpage to server 120. Server 120 deletes the shortpage and returns an indication to the client that the shortpage has been deleted. The server removes the name of the shortpage from its list of shortpages 332.

If the client wants to edit options for a shortpage, the client sends the URL of the shortpage and a request to edit the options of the shortpage. The client receives the edit page from server and displays it. The user then edits the options and sends the edits to the server.

Fig. 4(c) shows an overview of an exemplary shortpage method on the server side. When the server receives a request from the client's browser to create, edit, delete, or view a shortpage or option (element 450), the server translates the request into the appropriate request(s) to the web server(s) and retrieves the requested web page(s) or site(s) from the server(s) (element 452). In the described embodiment, these requests are HTTP requests and the web pages are written in HTML or a similar hypertext language. The server does the user-specified action (such as adding selection-enabling information when a shortpage is to be created or edited) or selecting the relevant parts of the web page (when a shortpage is to be viewed) (element 456). For example, the server adds the selection-enabling information to the retrieved web page and sends the resulting page to the client's browser. Table 1 shows an example of Web Procedure Calls (WPCs) used to communicate between the client and the server. The WPC is part of the URL sent by the client and has the format:

/UserName=<username>&ProcName=<procname>{&arg1=val1}{&arg2=val2}...{&arg last=vallast}

where a string in pointed brackets "<" ">" indicates a type of data and a string in curly brackets "{" "}" indicates optional arguments.

II. Creating/Editing a Shortpage

The following paragraphs provide exemplary details of how to create/edit a shortpage. The example shown in based on an exemplary web page shown in Fig. 5(a). The web page shown has a URL of http://abc.gov. In the example, a user visits this web site frequently and desires to select sub-elements of the web site to be placed on a shortpage. If the user desires to create a shortpage based on the web page of Fig. 5(a), he enters the URL of the web page into address 318 of Fig. 3 and selects "Create Shortpage" button 320. The server will retrieve the requested web page and add selection-enabling information to the page.

Figs. 5(b) and 5(c) show the web page of Fig. 5(a) displayed along with selection-enabling information. The server has broken the web page into blocks, each block having selection-enabling information. In the described embodiment, the selection-enabling information is a pair of show/hide boxes, having a check mark and an "X" respectively. The check mark indicates that the corresponding block will be shown in the shortpage. The "X" indicates that the corresponding block will be hidden (not shown) in the shortpage. Fig. 5(b) shows an editing area 510 and a preview area 520. Each of these areas can be sized and scrolled by the user. Initially all blocks in the page are marked as "hide." The example of Fig. 5(b) shows all blocks marked as "show." Therefore, in the example, preview area 520 shows all blocks in the page. In contrast, in Fig. 5(c), the user has clicked on the "X" box 505, causing the corresponding block 504 to be hidden in the shortpage. Note that the block 504 is not shown in the preview area 520 of Fig. 5(c).

Fig. 5(d) shows the web page of Fig. 5(a) with a different level of block detail. When the user clicks "more details" link 561 of Fig. 5(c), a command is sent to the server and the server re-determines the blocks of the page using a higher level of detail than previously used. (A similar "less detail" button 561' is displayed on the web page of Fig. 5(d)). A method of determining blocks on a page in accordance with a current level of detail is described below in connection with Fig. 9. In Fig. 5(d), for example, the page is broken into fewer blocks. For example, block 504 does not exist in editing area 510.

Fig. 5(e) shows a web page that allows the user to view/edit shortpage properties. The web page of Fig. 5(e) is requested when the user clicks on "Next" link 560, as shown in Figs. 5(b) - 5(d). This page allow the user to enter properties of a shortpage. The user can change the name associated with the shortpage (for example, the name shown in area 332 of Fig. 3) by entering a new name in area 570. The user can enter comments in area 572. The user can indicate that the shortpage does not show the page background in area 574. The user can indicate that changes since the last time the shortpage was viewed should be highlighted on the shortpage in area 576. The user can indicate that the shortpage should be used for similar pages and all URLs starting with a specified string (areas 578 and 580). Other properties could be included or certain properties shown in the figure excluded without departing from the spirit or scope of the invention. When the user clicks a "Done" button 582, the client sends the entered information to server 120, which stores the entered information in connection with the shortpage.

Fig. 6 is a flow chart showing how the client and the server interact to allow the user to create/edit a shortpage. Each element of the flow chart has a notation next to it indicating whether it is preferably performed by the client, server, or a combination. In element 602, after the server has retrieved a page from a web server, the server needs to break the page down into blocks and to add selection-enabling information to the page for each block. The server uses the SplitPage method (Fig. 8) and the Level of Detail method (Fig. 9) to break the page into blocks.

In element 604, the server adds selection-enabling information to the retrieved web page. Specifically, the server adds a "Show" button and a "Hide" button next to the block and puts a border (<table>) around it. The client's browser, thus, displays the web page along with the selection-enabling information for each block.

When the user selects "Show" or "Hide," an HTTP request that uniquely identifies the button is sent to the server (see Table 1). In element 608, the server marks the corresponding block as either "shown" or "hidden" in the representation tree (RepTree) for the shortpage. This information is a part of the selection information stored on the server. In element 610, the server returns a preview of the resulting shortpage to the user, which is displayed in a separate frame (preview area 520). The marked-up RepTree is stored as the template for future references to this shortpage. Thus, the RepTree forms at least part of the selection information. It will be understood that, although the example shows certain actions being performed by the client or the server, other embodiments may implement the client or server to perform actions not shown as being performed by them. For example, the selection information could be stored on the client side.

Fig. 7 shows an example of parsing used by the server to create and edit shortpages. This figure shows the Split Page method. In the example, a page is represented by an HTML file. The HTML is parsed in a manner known to persons of ordinary skill in the art to yield a Representation Tree. The Representation Tree has complex/multipart tags (e.g., <ul>), simple tags (e.g., <img src=Img.gif>), and text nodes (e.g., item1). A simple tag is defined as a tag that has no children and a complex tag is defined as a tag that has children. Server 120 parses the HTML file into a RepTree and iterates down the tree in a depth-first way to mark each node in the tree according to whether it is a single block. The RepTree includes a flag for each block indicating whether it is shown or hidden. A detailed example of the format of a RepTree is shown in Table 2. This format is not to be taken in a limiting sense.

Fig. 8 shows an example of a split page method used to create/edit shortpages. As shown in Fig. 8, if a node corresponds to a simple tag that is an image, input, iframe, or object, it is marked as a single block. If a node corresponds to a complex tag of type <tr> and there are more than two rows or columns in the table, mark the node. If a node corresponds to a complex tag of type <form>, do not mark the node. If a node corresponds to a complex tag of other types, mark the node if it has any text. For other types of nodes, continue traversing the RepTree. For each complex node, walk down the list of its children. Any nodes between two single blocks become one complex block. To avoid creating extra blocks, if a node has only one child block, give the block status to the parent node instead.

Fig. 9 is a flow chart of how to determine a display level when creating shortpages. In the described embodiment, the level of detail is used to decrease the number of blocks displayed at a given time. This flow chart demonstrates how to work with two levels (high and low), but any number of levels can be supported. If the level of detail is low, element 904 determines whether the block's parent has less than L nodes below its children. L can be, for example, 5. If the block's parent has less than L nodes below its children, the block is shown as a separate block when the user creates or edits a shortpage. Otherwise, the block is not shown as a separate block. If the level of detail is high, element 906 determines whether the block has more than K nodes below its children. K can be, for example, 1. If the block has more than K nodes below its children, the block is shown as a separate block when the user creates a shortpage. Otherwise, the block is not shown as a separate block.

Fig. 10 shows the startpage of Fig. 3 after a shortpage has been created based on the web page of Fig. 5(a). Server 120 indicates to the client that a shortpage has been created and adds the name of the shortpage to area 332 of the user's startpage. The startpage having the name of the new shortpage is sent to the browser, where it is displayed.

Fig. 11 shows the shortpage created from the web page of Fig. 5(a). In the example, the user selected certain sub-elements of the page. Specifically, the user indicated that all but a leftmost block of the page is marked as hidden. Thus, the shortpage shows only the non-hidden block. The shortpage also includes a link to the user's startpage 1102, a link to a notification page 1104, a link to a help page 1106, a link to a page that allows the user to edit a shortpage 1105 and a link to the fullpage 1103 upon which the shortpage is based. Note that the user has indicated that the background of the original page is not to be shown in this shortpage (see Fig. 5(e)).

Fig. 12 shows a fullpage corresponding to the web page of Fig. 11. The fullpage is displayed when the user clicks on the fullpage link 1103 of Fig. 11. In contrast to the shortpage, the original page's background is shown, since it is part of the page. Even though all blocks of the fullpage are shown, the full page is fetched via server 120 and not directly from the web page server. This is shown by the URL of the fullpage:
http://shortwave.com:9999/UserName=JSmith/ProcName=B_GetPage&purpose=7&
txtURL=http://www.abc.gov .

In this example, the user is "Jsmith". The action is to get/fetch a page. "Purpose = 7" indicates that a full page should be fetched. The URL of the full page is http://www.abc.gov. Further example of parameters used to communicate between the client and the server are described in Table 1.

III. Viewing a ShortPage

Fig. 13 is a flow chart showing how the client and server interact to allow the user to view a shortpage. In the described embodiment, the user has selected a name of a shortpage in area 332 and selected "go" in area 330. It should be remembered that, although a shortpage may be based on a particular web page, that web page may have changed since the shortpage was created, either because content may have been changed, added, or deleted. It is necessary to parse the web page on which the shortpage is based (the target) and determine whether the blocks marked "show" in the template are still present in the target. The target page on which the shortpage is based is retrieved and parsed into a RepTree by the server. Each shortpage has a RepTree associated with it.

In element 1304, the server performs a "double traverse" method, recursively traversing down both the template and the target RepTrees, trying to match the nodes. If two nodes match by type and tag, assign the "Show" or "Hide" flags of the template node to the target node. If there is no match, start to look for the closest match. If, for example, we are currently on nodes Temp[I], Targ[J], then we are looking for matching Temp[K] and Targ[L] such that, for K and L, K-I+L-J is the smallest possible value.

After element 1308, any template nodes that are not matched were the nodes that have disappeared from the page. Any target nodes that were not matched were the nodes that have been added to the page. All target nodes with "Show" flags (i.e., all target blocks that matched a block in the template) are part of the shortpage. These blocks are sent to the requesting client. Other appropriate matching methods can also be used.

In the described embodiment, the user can enter the address of a web page to be shortened. For example, the user can enter a URL in area 318 and click on button 340. If the user has previously defined a shortpage for this URL or for a URL on this web site, the shortpage is displayed. Similarly, if a page has a link to another page within it and the link is selected, server 120 will check the new link to determine whether it is a shortpage.

Fig. 14 is a flow chart showing details of determining whether a URL is the URL of a shortpage. As described in element 1402, this method will be used when, instead of requesting a specific shortpage, the user enters or browses to a particular URL. In element 1404, for each shortpage that the user has, server 120 compares the URL, host name, document name, and CGI argument lists. The server does this for both a shortpage requested address and for the actual address. The requested address and the actual address may be different because of HTTP redirection.

If, in element 1406, the host name, document, name, and CGI parameter names match, it is a "clear match." If a clear match occurs, use this shortpage for customizing the request. If a clear match does not occur, control passes to element 1410. In element 1410, for each "approximate match" (which is defined as host name and document name match), calculate "match ranking." Step 1412 calculates match ranking as follows. For each matching RepNode of RepTree, add (1/(Node depth)) to the rank. For each mismatch, subtract the same amount. In step 1414, select the shortpage with the highest matching rank for customizing this request. If no match occurs, then show the full page.

IV. Creating/Editing/Viewing a Combopage

A described embodiment of the present invention allows the user to define a "combination page" (also called a "combopage"). A combopage is a page made up of more than one web page. In the described embodiment, a combopage is made up of a combination of shortpages. Other combopages are made up of information from more than one web page, but not necessarily from other shortpages.

Fig. 15(a) shows a display for creating a combopage. In the example provided, combopages are created by combining shortpages. Combopages can also be created in a manner similar to that of shortpages without explicitly creating shortpages first. The display of Fig. 15(a) is displayed by a browser of a client. An area 1502 contains the names of all shortpages created by the user. Selected shortpage area 1504 contains the names of all shortpages selected by the user to be in the combopage. In the example, no shortpages have been selected yet.

Arrow buttons 1506 cause a shortpage name to be added to or removed from area 1504. Each time the user clicks on one of buttons 1506, a message is sent to server 120, which returns a redrawn page and stores the change to the combopage. Up and Down buttons 1508 allow the user to change the position of a shortpage within a combopage. Each time the user clicks on one of buttons 1508, a message is sent to server 120, which returns a redrawn page and stores the change in position of the elements in the combopage.

Fig. 15(b) shows an example of a combopage. In the example, the combopage is made up of two short pages: a first shortpage 1552 and a second shortpage 1554. The combopage also includes a link to the user's startpage, a help link, and a link to the edit page (see Fig. 16(c)). It should be noted that the shortpages used to make the combopage can be from the same site or from different sites.

Fig. 16(a) is a flow chart for creating a combopage using the display of Fig. 15(a). Element 1602 waits for the user to select the shortpages for the combopage and to position the shortpages on the combopage. Element 1604 stores the identities of the user-selected shortpages and their positions.

Fig. 16(b) is a flow chart of a method performed by server 120 to display a previously created combopage. Element 1652 receives a request for a combopage from a user. For each shortpage that is part of the combopage, the server sends a request to the corresponding web site server. In element 1656, the server waits until all the requests come back from the web site servers or a time-out expires. In element 1658, the server creates an HTML table for the combopage and, in element 1650, sends the resulting HTML to the client.

Fig. 16(c) shows a web page that allows the user to view/edit combopage properties. The web page of Fig. 16(c) is preferably requested when the user clicks on "Next" link 1508, as shown in Fig. 15(a). The user can change the name associated with the combopage by entering a new name in area 590. The user can indicate that the contents of the combopage are to be enclosed in an HTML table in area 592. When the user clicks a "Done" button 582, the client sends the entered information to server 120, which stores the entered information in connection with the combopage and returns the user to the user's startpage.

V. Creating/Editing a Notification

The described embodiment of the invention allows the user to enable a "notification" operation for blocks (sub-elements) of a web page. When a specified block on a web page changes in a user-specified manner, the user is notified. It is important to note that the web page is broken into blocks in a manner similar to that discussed above for making shortpages. Each block of the page can have a notification set for that block. The notification properties for various blocks on a page can be different from each other. Some blocks can have their notifications enabled, while others can have their notifications disabled. Some can have different notification methods than other and/or can have different notification conditions than others on the same page. Notification can be turned on for some blocks of a page and turned off for other blocks on the page. Thus, the granularity of notifications is smaller than the page level.

Fig. 18(a) is a flow chart showing an example of how notifications are enabled. First, the server breaks the page into blocks as if it were creating a shortpage. The server adds notification-enabling information to the page. In the example shown, the notification-enabling information is an envelope icon, such as icons 1702, 1704 of Fig. 17(a). As shown in element 1806, when the user selects the block to be tracked (by clicking on the associated icon), a notification properties page is displayed (see Fig. 17(b)) and the user is asked to select notification properties, including: a notification condition, the frequency of checking notification, and the notification device. These properties are sent to the server, which sets a "notify" flag in the node of the RepTree corresponding to the block. The RepTree, along with the notification parameters are saved on the server as a template.

Fig. 17(a) shows a display that allows a user to enable notifications for the page shown in Fig. 5(a). The server has added notification-enabling information to the page. In the example shown, the notification enabling information includes a notification icon for each block on the page. Examples of notification icons include an icon 1702 and an icon 1704. The user has set a notification flag for the block corresponding to icon 1704, and server 120 has added a "Notify if changed" banner 1705 below the icon. Each icon is a link that causes the display of Fig. 17(b) to be displayed.

In at least one embodiment, a notification can be conditioned on more than one block or sub-element. The user is allowed to indicate a notification condition involving more than one block. For example, if there are two blocks on the page (block A and block B) that the user is interested in, he could indicate "notify me when block A contains 'new book' and block B < $30." The user could indicate this notification condition, for example via a pop-up window or other appropriate mechanism.

Fig. 17(b) shows a display that allows a user to edit notification properties. Each block on a page has separate notification properties. As shown in Fig. 17(b), the status of icon 1704 is enabled 1752. The name for the notification is chosen to be the same as the name of the web page (www.abc.gov) although the user could choose another name. The user has chosen to be notified of changes via e-mail 1758. The e-mail will have the subject "www.abc.gov" 1764. The user could also choose to be notified by a pager message 1760 or to have his notification displayed on a "notification" web page 1762. The user has indicated that he wants to be notified if the indicated page changes in any way 1766. Other possibilities for notification include: always, if contains (a specified value), if equal to, if not equal to, if greater than, if greater than or equal to, if less than, if less than or equal to. The user has indicated that the indicated web page should be monitored for changes every hour. If the specified change criteria are met, a notification is sent via the appropriate notification method (here, via e-mail). Monitoring can also occur in units of minutes, days, weeks, or months, or any other appropriate units. When the user selects the "back" 1770 button, the properties are sent to the server and the previous page is displayed. When the user selects the "done" 1772 button, the properties are sent to the server and the user's startpage is displayed.

Fig. 17(c) shows a notification page on which notifications of changes are displayed. In the example shown, no notifications have occurred since the page was last viewed. If changes meeting the specified criteria had occurred, a message would appear on this web page.

Fig. 18(b) is a flow chart of a method for issuing a notification. As shown in element 1852, the server waits for the specified time interval (or periodically checks to determine whether the specified time interval has passed). After the specified time interval has passed, the server retrieves the page having a notification enabled from the web site server. Using, for example, the same double traverse process described in Fig. 13, the server finds the target node (i.e., block on the page) that corresponds to the template node in the RepTree having its notify flag set. The block in the fetched page is checked to determine whether it meets the notification condition specified by the user (for example, has this block changed). If the notification condition is met, the server sends a notification message to the user via the mechanism specified by the user (for example, an e-mail message, a pager message, or an entry on a notification web page). If the condition is not met, the server continues with the next notification in the RepTree. Different blocks (sub-elements) on a page can have different notification frequencies.

VI. Handheld client

As discussed in connection with Fig. 1, the present invention can be implemented for a wide variety of clients. Fig. 20 is a block diagram of data flow when a shortpage or a combopage is viewed on a personal digital assistant, such as a Palm VII Personal Digital Assistant (Available from 3Com Corporation) or a similar handheld device. PDAs often use a "web clipping" system to view web pages. The PDA views predefined web clippings by sending a request for the clipped pages to a proxy server.

Initially, a "web clipping" application, which contains a single link to the user's startpage is created and downloaded into the user's handheld device. The user creates his shortpages and combopages as described above. Preferably, these shortpages and combopages are created on the user's PC, although other embodiments may allow the user to create them on the handheld.

To view a shortpage or a combopage, the user simply goes to his startpage "web clipping" in the handheld and selects a link to the shortpage. As shown in Fig. 20, this action causes a request to be sent to the proxy server 2004, which passes the request to the shortpage server 2006, which forwards the request to the web site 2008 hosting the full page (of full pages in the case of a combopage). The web site returns the requested page(s) to the shortpage server 2006, which creates a shortpage or a combopage as described above and which then passes the created shortpage or combopage to the proxy server 2004. The proxy server sends the shortpage or combo page to the handheld client 2002.

Fig. 19 is a flow chart for viewing a shortpage when the client is a personal digital assistant. In element 1902, a web clipping short page application, which contains a single link to the user's startpage is created and downloaded into the user's PDA. In element 1904, the user creates his shortpage (or combopage) as before, using his PC or similar device. In element 1906, to view a shortpage or combopage, the user simply goes to his startpage "web clipping" and selects a link to the shortpage. In element 1908, the request goes through a proxy server (such as palm.net of 3Com) and is forwarded to the site hosting the page or pages. In element 1910, the response from the web server is customized by the Shortpage server and goes back to the user.

In the described embodiment, the shortpage server creates a modified version of a shortpage when it knows that the client is a handheld device. For example, the shortpage server removes large graphic files and Java Script code. Other embodiments may make other modifications or no modifications, depending on the nature and capabilities of the handheld client. In the described embodiment, the proxy server also modifies the page somewhat before sending the page to the handheld client.

VII. Shared Portal

Fig. 21 is a display of a shared portal shortpage application. A shared portal is created by one or more persons and is viewable by multiple persons. Different users may have different access permissions stored within the server. For example, in one embodiment, a single user has permissions to create folders (such as folder 2102), but any user can add shortpages within folders (link 2104 is a link to a combopage titled "Bay Area News"). As another example, only a single user can add folders and links, but any user can access the shortpages and combopages. In a shared portal, whenever a user attempts to edit or create a shortpage or combopage, the server first checks the permissions of the user. The user may be required to enter a password and/or to Log on from an approved machine. Similarly, whenever a user attempts to access a shortpage or a combopage, the server checks the permissions of the user to determine whether the user is authorized to access the page he has requested. A shared portals may be open to all users or may be restricted to certain groups, such as families, employees of a company, employees of a department, or clubs.

VIII. Cluster Implementation

In one embodiment of the present invention, server 120 resides on a number of nodes, which are dual-Pentium Windows NT PCs. In this implementation, data is stored in a shared RAID file server. A hardware load-balancer, such as a Big/IP load balancer from FS Networks, Inc. of Seattle, Washington is used to route user requests to the least busy node. Such an implementation uses persistent connections with, for example, a ten minute timeout to ensure that the requests from the same address get to the same server. When the first request for the given user comes to the given server, the user's data is read from the file server. If there is no activity for the time Q (for example, Q>T, 12 minutes), the data for the user is cleared from memory.

While the invention has been described in conjunction with a specific embodiment, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art in light of the foregoing description. For example, in at least one implementation, the nodes handling notification are kept separate from shortpage/combopage machines, since realtime response is not typically critical for notification. In another embodiment, shortpages and/or combopages are pre-created and offered to third persons who want specialized information or who want shorter web pages. In another embodiment, shortpages can be located on a combopage next to each other in a horizontal direction. Accordingly, it is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims and equivalents.

Table 1 (Page 1 of 1)

WPCs - Web Procedure Call list

All WPCs take a username.

```
// The WPC consists of: (without "<" or ">")
//
/Username=<username>&ProcName=<procname>&arg1=val1&arg2=val2&...&argLast=valLas
t
// (The number of args is variable, and the "last" arg is optional).


// New user registration and user login.
NewUserLogin - Allows the user to enter registration information
RegisterUser - Submits username, full name, password, email, and other info for a new user.
LoginUser - login user. Takes username and password

// Wizard Action
WizardAction - takes object (shortpage, notification, or combopage) id
        Edit            - start editing the object
        Props           - Displays properties of the object
        CreateShortpage - creates a shortpage
        CreateNotification - creates a notification
        CreateCombopage - creates a combopage
        Browse - browse to the given URL
        Go - go to the given shortpage/combopage
        Delete - delete given object
        Next - present logically the next screen in the wizard
        Done - done with the editing
        Cancel - cancel editing

B_GetPage - Get a page. Takes a URL
B_GetPageS - Same, but the request came through a script, URL needs some massaging. Takes
a URL and a base URL for the page to be appended to it.
StartPage - show Start Page for the user
NotifPage - Show notification page for the user
U_DeleteNotifs - delete certain notification messages. Takes message ids.
P_SelectionFrame - Displays the selection frame (upper frame) for the shortpage creation
P_DoCommand - do editing command, such as "Show" block, or "Hide" block.
        Takes block id as a parameter, and command id ("Show", "Hide", "Notify", etc.)

ProfileMgrAction - Edit profile settings
```

Table 2 (Page 1 of 2)

```
typedef CArray<CRepNode*, CRepNode*> CRepNodeArray;

class CRepNode
{
        friend class CRepBuilder;
public:
        CRepNode(CRepNode *pParent=NULL, int iIndex=0);
        virtual ~CRepNode();

        void AddChild(CRepNode *pChild);
        void InsertChild(CRepNode *pChild, int iIndex);

        void Serialize(CMyArchive& ar);

        enum Type{
                NONE,
                TEXT,
                SIMPLE_TAG,
                CONTAINER_TAG,
                COMMENT,
                BADTAG,    // bad closing tag.
                SCRIPT,    // script node contains both <SCRIPT> and </SCRIPT>
                WHITESPACE,// whatever "trims" to an empty string.
                BAD_CLOSING
        };

        Type m_Type;

#define FLG_BLOCK_BEGIN  0x1
#define FLG_BLOCK_END  0x2

#define FLG_BLOCK_MASK  (FLG_BLOCK_BEGIN | FLG_BLOCK_END)

#define FLG_HIDE           0x10 // hide this node.
#define FLG_CHANGED                0x20 // node has changed
#define FLG_NOMATCH                0x40 // couldn't match this node.
#define FLG_SHOW           0x80 // show the node and subnodes.

#define FLG_NOTIFY                 0x100 // Notify on this node.
#define FLG_BADLY_CLOSED 0x200 // This node is incorrectly nested.
#define FLG_TEMP           0xf000 // The mask for temp bits used by stuff
                                            // here and there for a short time.
```

Table 2 (Page 2 of 2)

```
        ULONG m_Flags;
        CNodeInfo *m_pInfo; // any data can be stored here.
                            // not saved to the disk!

        CNotifyInfo *m_pNotify; // Notification info on this node.
                                            // must be serialized.

        // has either the text, or a simple tag or
        // the full tag with params, but without nested tags.. Example:
        // "<TD WIDTH=117 VALIGN=TOP ALIGN=LEFT>"
        CString m_Text;

        // Children of a complex tag.
        CRepNodeArray m_aKids;


        // the following two fields are redundant for all known (to our
        // program) tags.

        TAG m_Tag;
        // The tag without params and < or >, like "TD"
        // Note that there is never a closing tag, like </TD> in the representation.
        // If the type of the node is CONTAINER_TAG, then it is assumed that
        // there is a closing tag after all m_aKids.
        CString m_TagName;

        CString m_NodeHeader;

        CRepNode *GetParent() { return m_pParent; }
        int GetIndex() { return m_Index; }

        CRepNode *m_pParent; // root object has this NULL
        int m_Index;    // index into the array of the parent node.

#if DEBUG
        void AssertValid(); // CObject-like validation function.
#endif
};
```

WHAT IS CLAIMED IS:

1. A method to notify a user of certain changes to a web page, comprising:

allowing a user to choose sub-elements on the web page as subjects of notification;

saving the user's choices; and

monitoring the user's chosen sub-elements on the web page and notifying the user when a notification condition is true for at least one of the sub-elements.

2. The method of claim 1, further comprising splitting the web page into sub-elements before allowing the user to choose the sub-elements.

3. The method of claim 1, further comprising adding selection-enabling information to the web page to enable the user to choose the sub-elements.

4. The method of claim 1, further comprising:

notifying the user, via a user-selected notification mechanism, that the notification condition has been met for a sub-element.

5. The method of claim 1, further comprising:

notifying the user by e-mail.

6. The method of claim 1, further comprising:

notifying the user by placing a notification on a notification web page.

7. The method of claim 1, further comprising:

notifying the user by sending a pager message to the user.

8. The method of claim 1, further comprising:

notifying the user by displaying a notification on a client system.

9. The method of claim 8, wherein the client system is a personal computer.

10. The method of claim 8, wherein the client system is a personal digital assistant device.

11. The method of claim 8, wherein the client system is a web-enabled telephone having a display.

12. The method of claim 8, wherein the client system is an alphanumeric paging device.

13. The method of claim 8, wherein the client system is a web-enabled device

14. The method of claim 1, wherein the user can explicitly disable notification for a sub-element.

15. The method of claim 1, wherein the user can provide a name for the notification.

16. The method of claim 1, wherein the notification condition includes a frequency for how often the notification condition should be checked.

17. The method of claim 1, wherein the notification condition includes a check to see whether any change has occurred to the sub-element.

18. The method of claim 1, wherein the notification condition indicates to always notify the user.

19. The method of claim 1, wherein the notification condition includes a check to see whether the sub-element contains a specified value.

20. The method of claim 1, wherein the notification condition includes a check to see whether the sub-element has at least one of the following relations to a specified value: equal to, greater than, and less than.

21. The method of claim 1, wherein the user is allowed to indicate a notification condition involving more than one block.

22. An apparatus that notifies a user of certain changes to a web page, comprising:

a software portion configured to allow a user to choose sub-elements on the web page as subjects of notification;

a software portion configured to save the user's choices; and

a software portion configured to monitor the user's chosen sub-elements on the web page and to notify the user when a notification condition is true for at least one of the sub-elements.

23. The apparatus of claim 22, further comprising a software portion configured to split the web page into sub-elements before allowing the user to choose the sub-elements.

24. The apparatus of claim 22, further comprising a software portion configured to add selection-enabling information to the web page to enable the user to choose the sub-elements.

25. The apparatus of claim 22, further comprising:

a software portion configured to notify the user, via a user-selected notification mechanism, that the notification condition has been met for a sub-element.

26. A computer program product, comprising:

a computer readable medium having instructions thereon to notify a user of certain changes to a web page, including:

computer program devices configured to allow a user to choose sub-elements on a web page as subjects of notification;

computer program devices configured to save the user's choices; and

computer program devices configured to monitor the user's chosen sub-elements on the web page and to notify the user when a notification condition is true for at least one of the sub-elements.

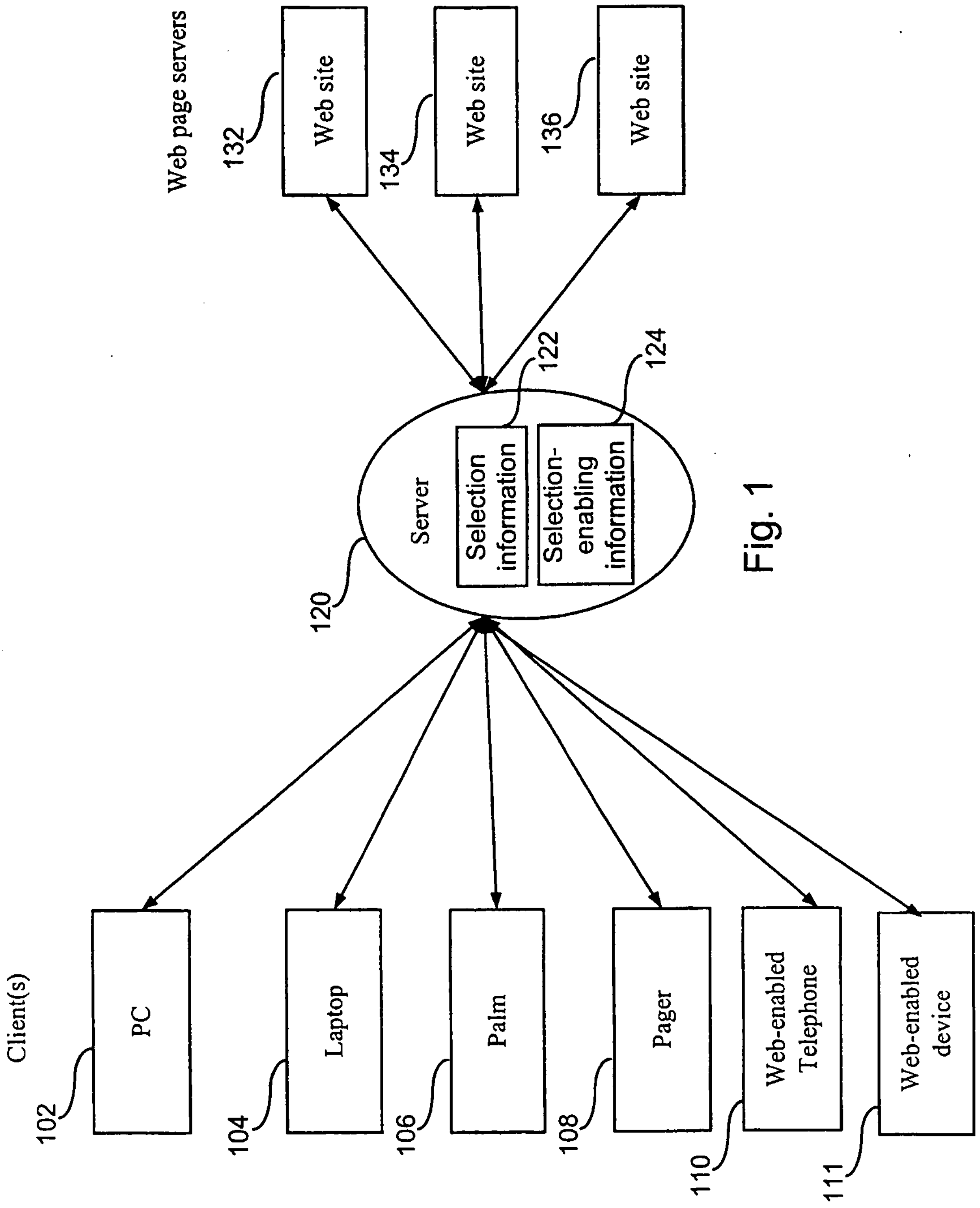

Web page servers
132
Web site
134
Web site
136
Web site
122
124
Server
Selection information
Selection-enabling information
120
Client(s)
102
PC
104
Laptop
106
Palm
108
Pager
110
Web-enabled Telephone
111
Web-enabled device

Fig. 1

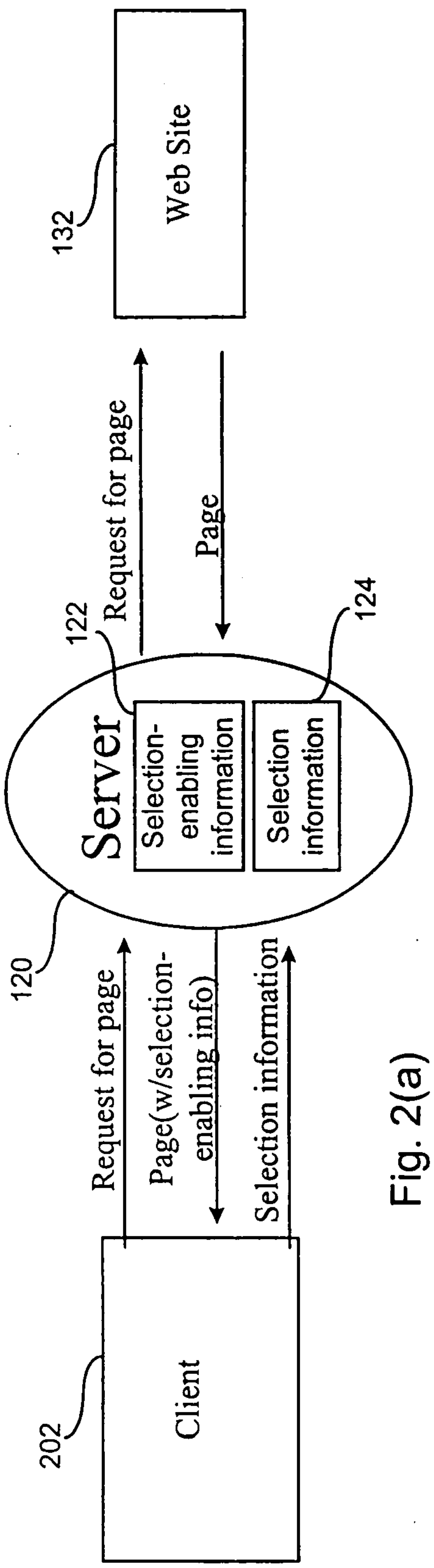


Fig. 2(a)
Creating a ShortPage

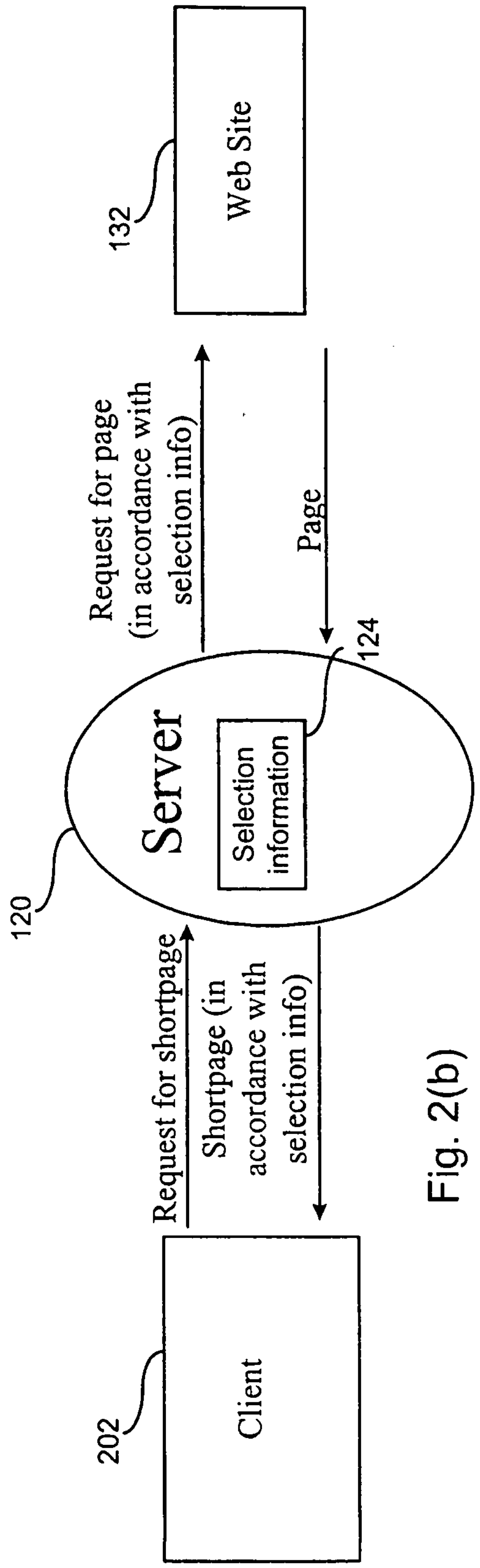

132
Web Site
Request for page (in accordance with selection info)
Page
124
Server
Selection information
120
Request for shortpage
Shortpage (in accordance with selection info)
202
Client

Fig. 2(b)
Viewing a ShortPage

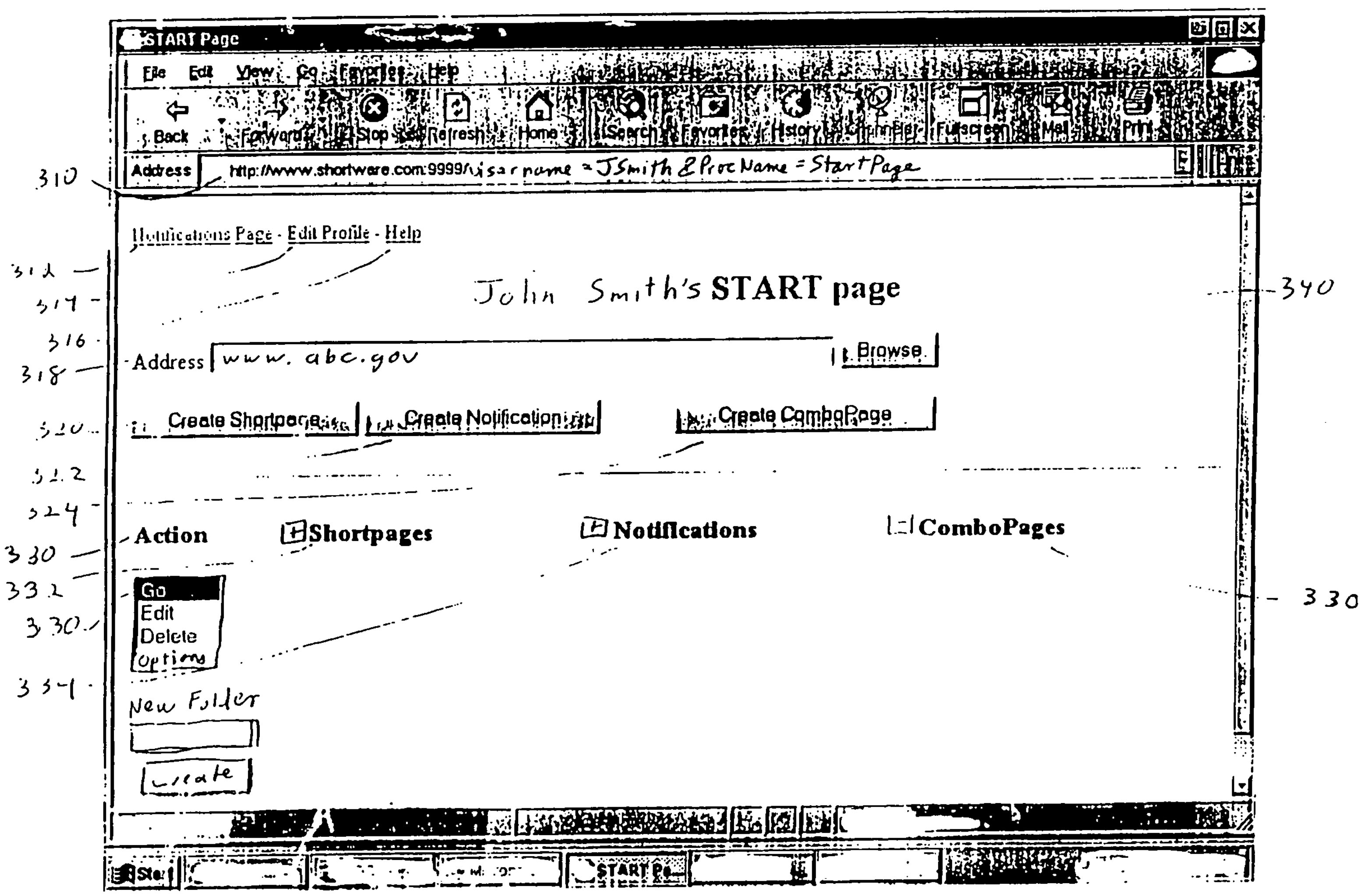


Fig. 3(a)
Before Creating a Shortpage

```
<html>

<head>
<title>START Page</title>
</head>
<script Language="JavaScript"> function ButtonClicked(buttonName)
 {
 if(document.main_form.txtURL.value == '')
 {
 alert('Please enter a URL.');
 return false;
 }
 else
 {
 document.main_form.btnAction.value = buttonName;
 return true;
 }
 }
 </script>


<body>
<font size="-1" color="#2244ff">

<p><font size="-1'><A href="http://shortpage.com:9999/UserName=lma
jerus&ProcName=NotifPage" target="_top"><font color="#2244ff">Noti
fications Page</font></A> - <A href="http://shortpage.com:9999/Use
rName=lmajerus&ProcName=START_F" target="_top"><font color="#2244f
f">Edit Profile</font></A> - <A href="http://shortpage.com:9999/Us
erName=lmajerus&ProcName=Help&Index=3&Browser=1" target="_top"><fo
nt color="#2244ff">Help</font></A></font>
<!-- END OF NAVIGATION BAR -->

</font> </p>

<h2 align="center">lmajerus's START page</h2>
<font size="-1" color="#0000FF">

<p></font></p>

<form action="http://shortpage.com:9999/UserName=lmajerus&ProcName
=WizardAction&Place=Start" method="POST" name="main_form"
onSubmit="if(txtURL.value == ''){        alert('Please enter a URL.
```

Fig 3(b)

```
');     return false;}else if(btnAction.value == 'invalid'){     bt
nAction.value = 'Browse';        return true;}">
  <input type="hidden" name="btnAction" value="invalid"><p>Address
 <input type="text"
  size="60" name="txtURL" value>&nbsp; <input type="submit" name="
btnl" value="Browse"
  onClick="return ButtonClicked('Browse');"></p>
  <p><input type="submit" name="btn" value="Create Shortpage"
  onClick="return ButtonClicked('Create Shortpage');"> <input type
="submit" name="btn"
  value="Create Notification" onClick="return ButtonClicked('Creat
e Notification');">
  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;<input
  type="submit" name="btn" value="Create ComboPage"
  onClick="btnAction.value = &quot;Create ComboPage&quot;; submit(
); return false;"> </p>
  <hr>
</form>
<script Language="JavaScript">
function DoAction(type, id)
 {
 if(document.second_form.btnAction[document.second_form.btnAction.
selectedIndex].value == 'Delete')
 {
  doDelete = confirm('Delete this ' + type + '?');
  if(doDelete)
  {
   document.second_form.txtID.value = id;
   document.second_form.submit();
  }
 }
 else
 {
  document.second_form.txtID.value = id;
  document.second_form.submit();
 }
 return false;
 }
 </script>


 <form action="http://shortpage.com:9999/UserName=lmajerus&ProcName
```

Fig. 3(c)

```
=WizardAction&Place=Start" method="GET" name="second_form">
  <input type="hidden" name="txtID" value><table width="100%">
    <tr valign="top">
      <td width="16%"><h3>Action</h3>
      <p><select name="btnAction" size="4">
        <option value="Go" selected>Go</option>
        <option value="Edit">Edit</option>
        <option value="Options">Options</option>
        <option value="Delete">Delete</option>
      </select><!-- This is not a button but using the same name m
akes things very convenient --><br>
      </p>
      <p>New Folder:<br>
      <input type="text" name="NewFolderName" size="9"> <br>
      <input type="submit" value="Create!" name="btnNewFolder"></p
>
      <p>&nbsp;</td>
      <td width="84%"><table width=100%><tr valign=top>
<td width=33%><a href=http://shortpage.com:9999/UserName=lmajerus&
ProcName=FolderOp&OpCode=Collapse&FolderID=116761&LoopBack=http%3A
%2F%2Fshortpage.com%3A9999%2FUserName%3Dlmajerus%26ProcName%3DStar
tPage><img src=http://www.shortware.com/images/minus_button.gif bo
rder=0></a><big><font color="#009900">ShortPages</font></big><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89115" onClick='return DoAction("ShortPage", 89115);'>
<font color="#2244ff">www.uspto.gov</font></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89117" onClick='return DoAction("ShortPage", 89117);'>
<font color="#2244ff">www.uspto.gov/cgi-bin/phone/ptophone</font><
/A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89126" onClick='return DoAction("ShortPage", 89126);'>
<font color="#2244ff">164.195.100.11/netahtml/search-bool.html</fo
nt></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89128" onClick='return DoAction("ShortPage", 89128);'>
<font color="#2244ff">www.uspto.gov</font></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89129" onClick='return DoAction("ShortPage", 89129);'>
<font color="#2244ff">www.uspto.gov</font></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89130" onClick='return DoAction("ShortPage", 89130);'>
<font color="#2244ff">www.uspto.gov</font></A><BR>
```

Fig. 3(d)

```
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89133" onClick='return DoAction("ShortPage", 89133);'>
<font color="#2244ff">www.optipat.com</font></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=111525" onClick='return DoAction("ShortPage", 111525);
'><font color="#2244ff">www.uspto.gov</font></A><BR>
<BR>
<a href=http://shortpage.com:9999/UserName=lmajerus&ProcName=Folde
rOp&OpCode=Expand&FolderID=116765&LoopBack=http%3A%2F%2Fshortpage.
com%3A9999%2FUserName%3Dlmajerus%26ProcName%3DStartPage><img src=h
ttp://www.shortware.com/images/add_button.gif border=0></a><big><f
ont color="#009900">test</font></big><BR>
<BR>
</td>
<td width=33%><a href=http://shortpage.com:9999/UserName=lmajerus&
ProcName=FolderOp&OpCode=Collapse&FolderID=116762&LoopBack=http%3A
%2F%2Fshortpage.com%3A9999%2FUserName%3Dlmajerus%26ProcName%3DStar
tPage><img src=http://www.shortware.com/images/minus_button.gif bo
rder=0></a><big><font color="#009900">ComboPages</font></big><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89118" onClick='return DoAction("ComboPage", 89118);'>
<font color="#2244ff">ComboPage #1</font></A><BR>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=103560" onClick='return DoAction("ComboPage", 103560);
'><font color="#2244ff"></font></A><BR>
<BR>
</td>
<td width=33%><a href=http://shortpage.com:9999/UserName=lmajerus&
ProcName=FolderOp&OpCode=Collapse&FolderID=116763&LoopBack=http%3A
%2F%2Fshortpage.com%3A9999%2FUserName%3Dlmajerus%26ProcName%3DStar
tPage><img src=http://www.shortware.com/images/minus_button.gif bo
rder=0></a><big><font color="#009900">Notifications</font></big><B
R>
<A href="http://shortpage.com:9999/UserName=lmajerus&ProcName=P_Ge
tPagelet&ID=89111" onClick='return DoAction("Notification", 89111)
;'><font color="#2244ff">www.uspto.gov</font></A><BR>
<BR>
</td>
</tr></table>
</td>
      </tr>
    </table>
</form>
</body>
</html>
```

Fig 3(e)

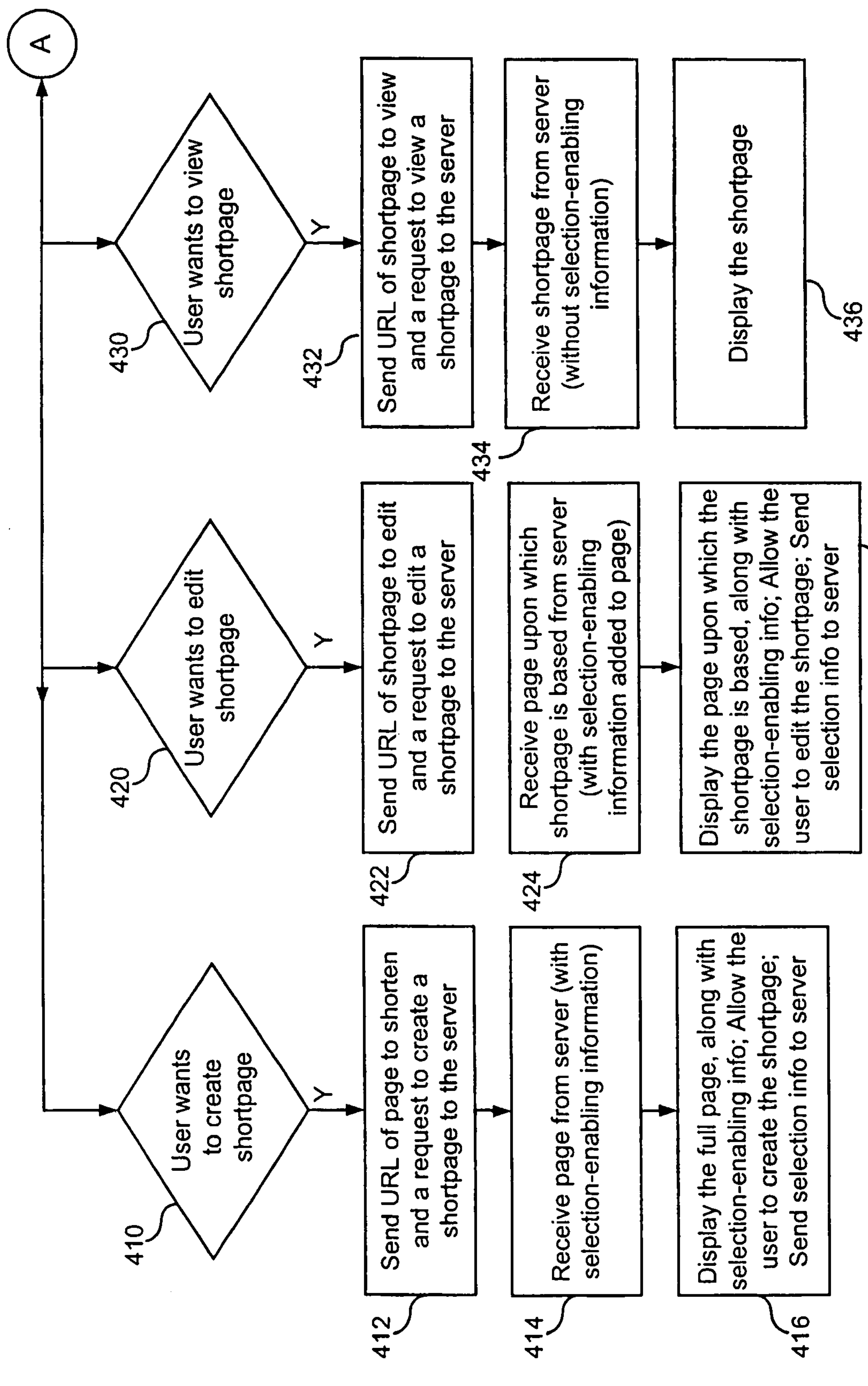

A
410
User wants to create shortpage
Y
412
Send URL of page to shorten and a request to create a shortpage to the server
414
Receive page from server (with selection-enabling information)
416
Display the full page, along with selection-enabling info; Allow the user to create the shortpage; Send selection info to server
420
User wants to edit shortpage
Y
422
Send URL of shortpage to edit and a request to edit a shortpage to the server
424
Receive page upon which shortpage is based from server (with selection-enabling information added to page)
426
Display the page upon which the shortpage is based, along with selection-enabling info; Allow the user to edit the shortpage; Send selection info to server
430
User wants to view shortpage
Y
432
Send URL of shortpage to view and a request to view a shortpage to the server
434
Receive shortpage from server (without selection-enabling information)
436
Display the shortpage

Fig. 4(a)
Shortpage Operations (client side)

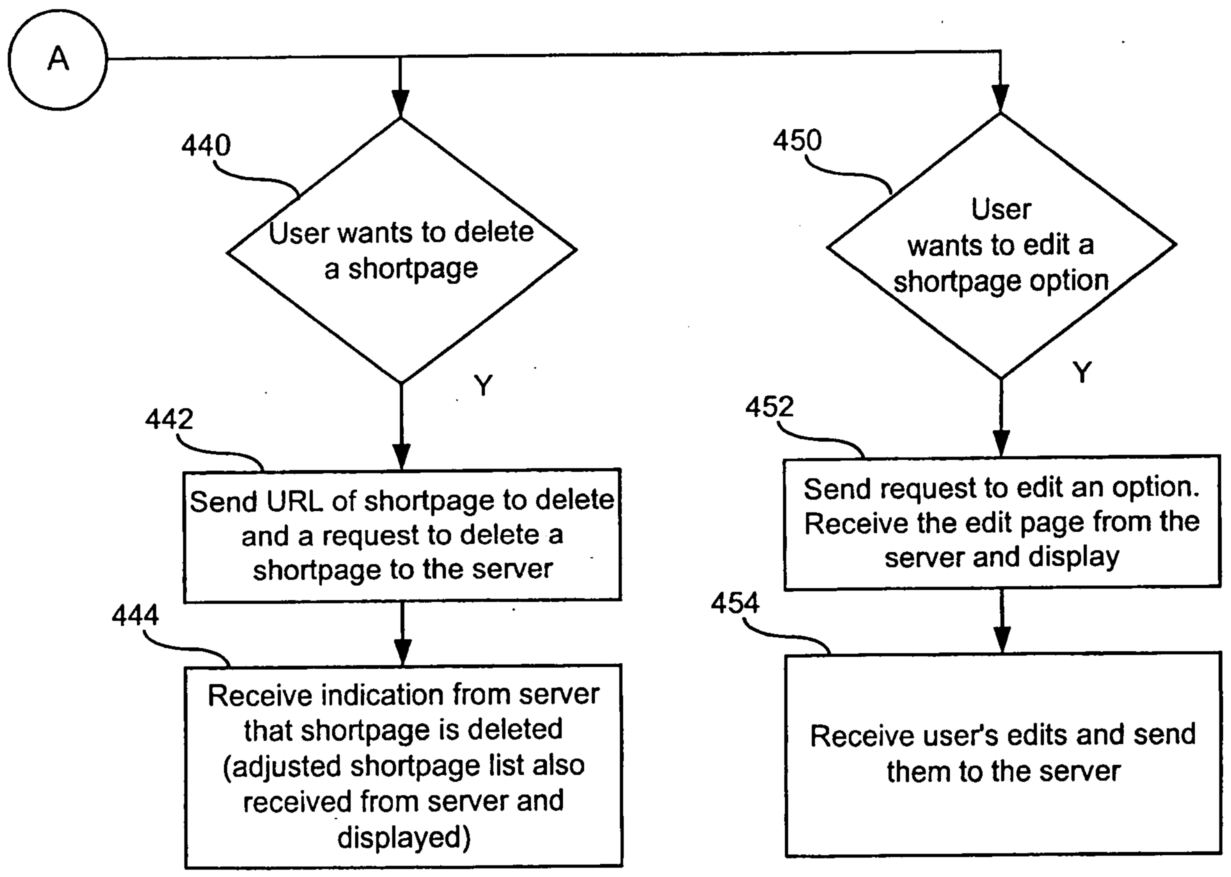

A
440
User wants to delete a shortpage
Y
442
Send URL of shortpage to delete and a request to delete a shortpage to the server
444
Receive indication from server that shortpage is deleted (adjusted shortpage list also received from server and displayed)
450
User wants to edit a shortpage option
Y
452
Send request to edit an option. Receive the edit page from the server and display
454
Receive user's edits and send them to the server

Fig. 4(b)
Shortpage Operations
(Client Side)

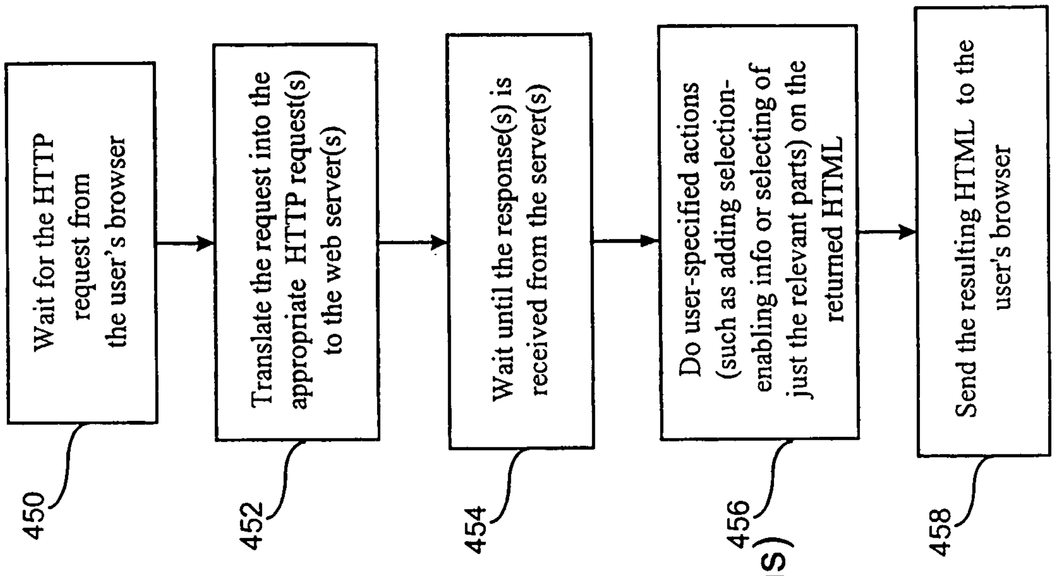
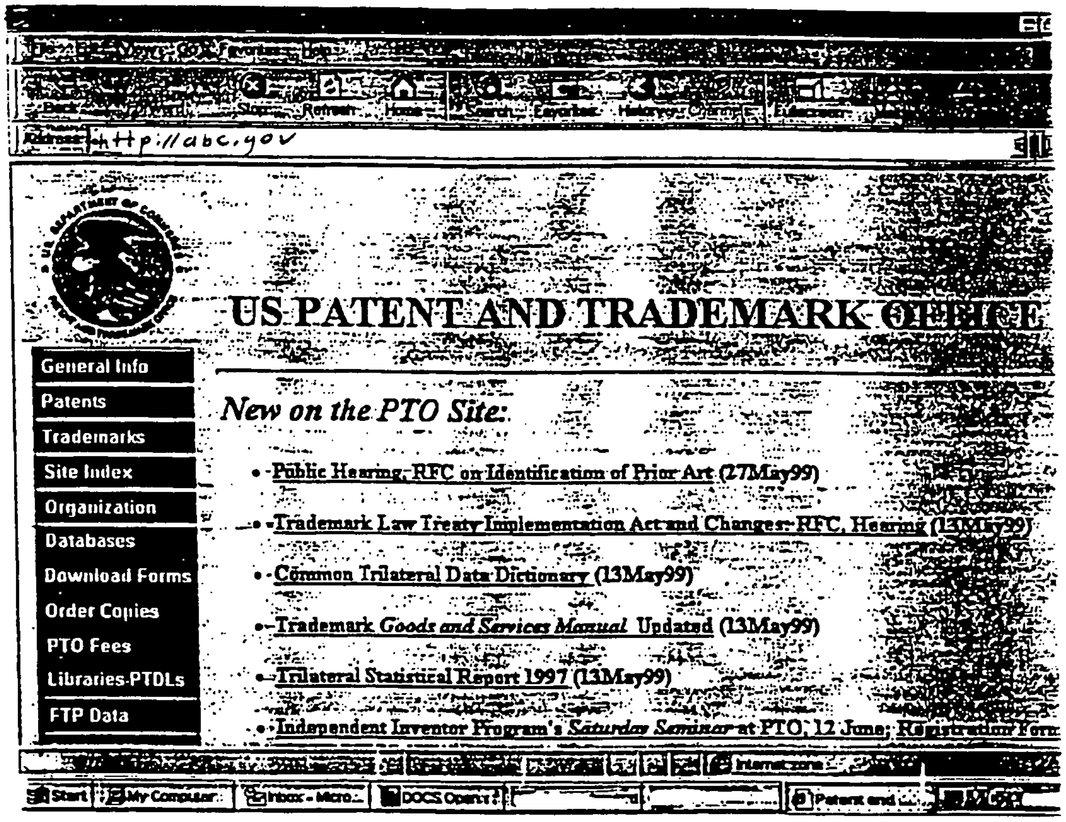

Wait for the HTTP request from the user's browser
Translate the request into the appropriate HTTP request(s) to the web server(s)
Wait until the response(s) is received from the server(s)
Do user-specified actions (such as adding selection-enabling info or selecting of just the relevant parts) on the returned HTML
Send the resulting HTML to the user's browser
450
452
454
456
458

Fig. 4(c)
Shortpage Operations
(overview of server actions)



US PATENT AND TRADEMARK OFFICE

New on the PTO Site:

- Public Hearing, RFC on Identification of Prior Art (27May99)
- Trademark Law Treaty Implementation Act and Changes; RFC, Hearing (13May99)
- Common Trilateral Data Dictionary (13May99)
- Trademark *Goods and Services Manual* Updated (13May99)
- Trilateral Statistical Report 1997 (13May99)
- Independent Inventor Program's *Saturday Seminar* at PTO, 12 June; Registration Form

Fig 5(a)

A web page

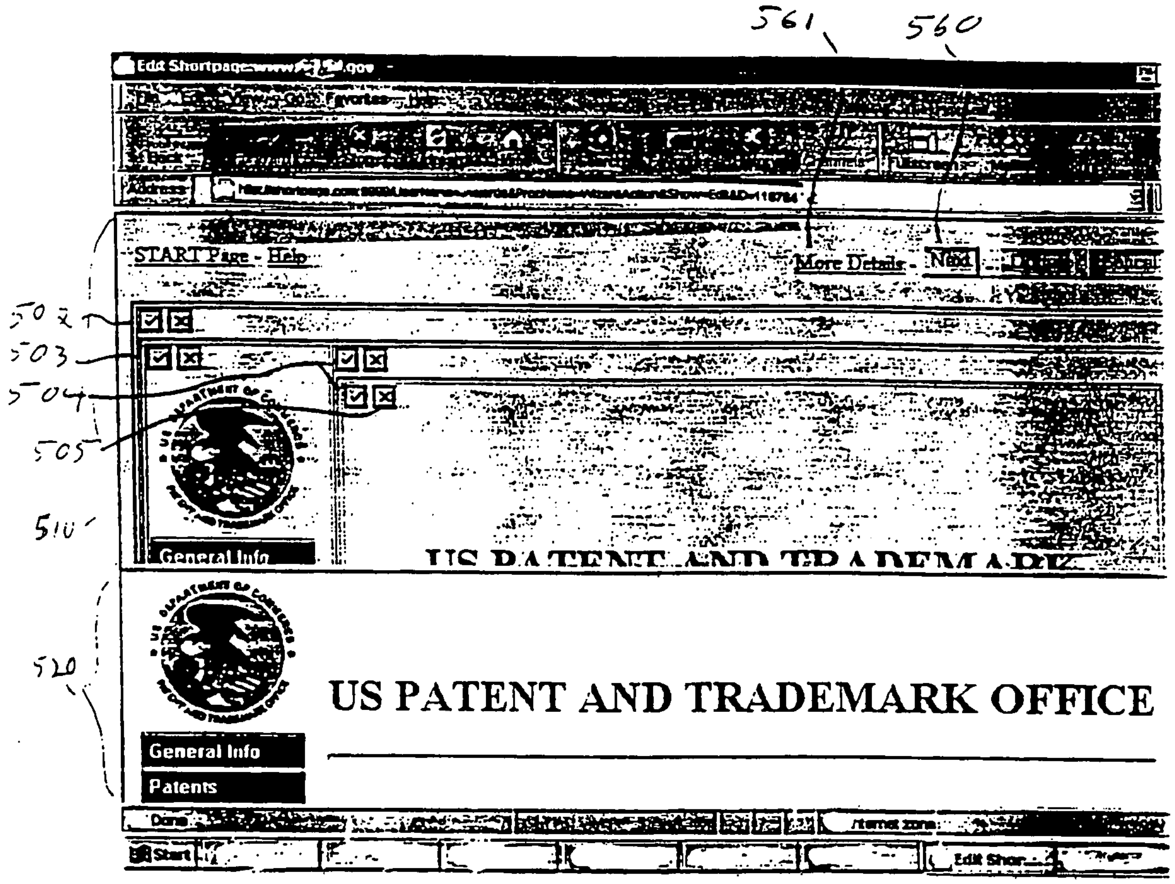


Fig. 5(b)
Create/Edit Shortpage

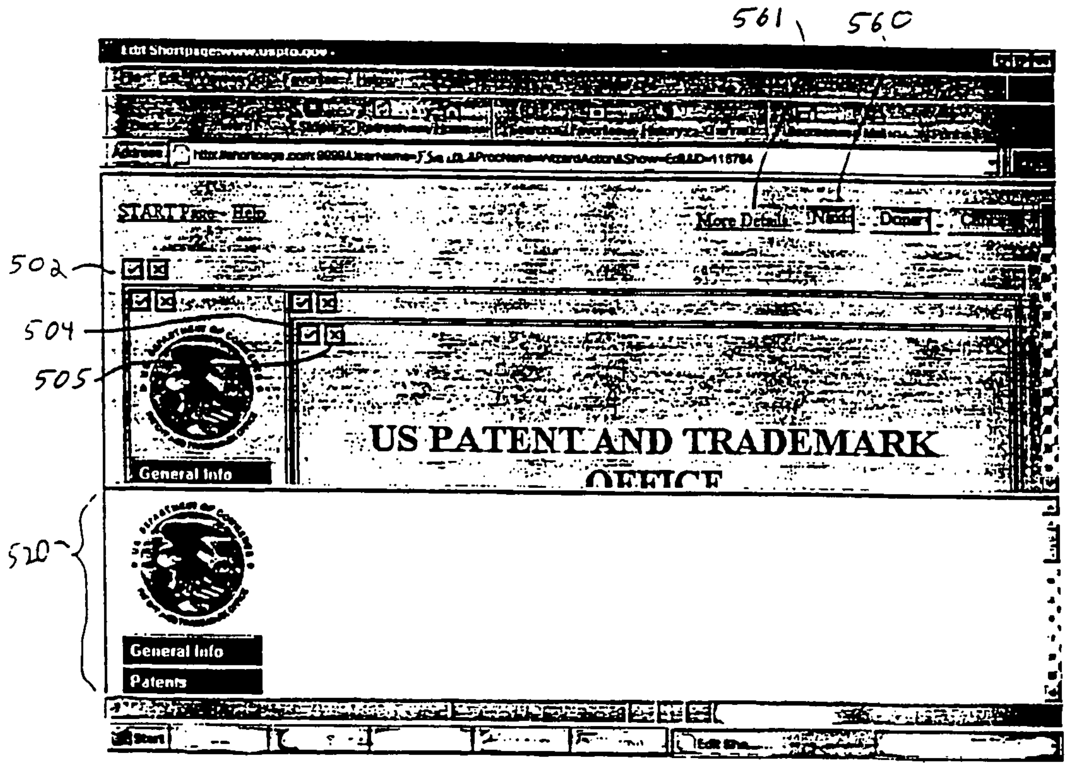


Fig 5(c)
Create/Edit Shortpage

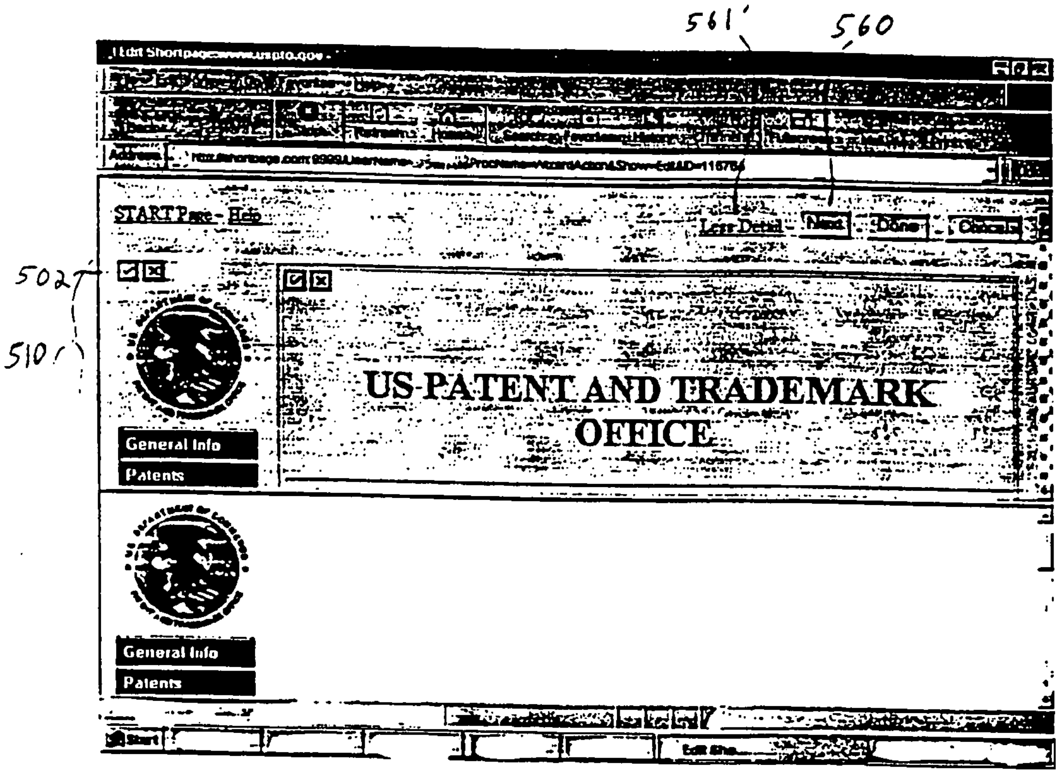
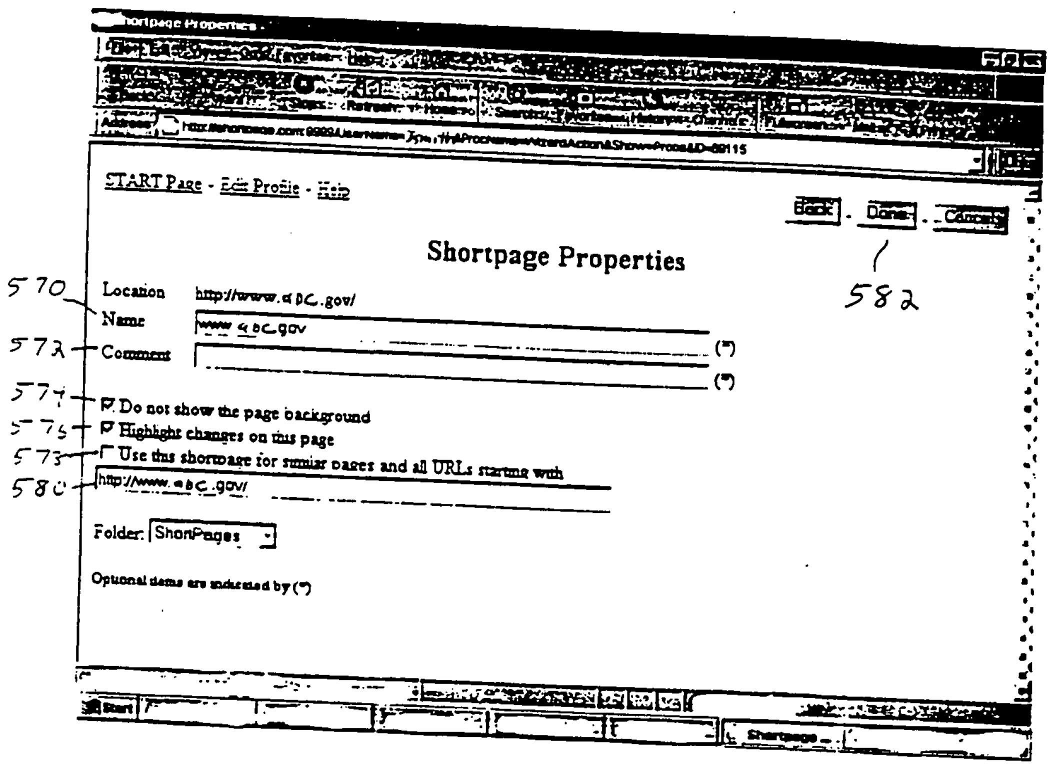


Fig. 5(d)
Create/Edit Shortpage



Shortpage Properties

Location http://www.abc.gov/
Name www.abc.gov
Comment (*)
(*)
☑ Do not show the page background
☑ Highlight changes on this page
☐ Use this shortpage for similar pages and all URLs starting with
http://www.abc.gov/
Folder: ShortPages
Optional items are indicated by (*)

Fig 5(2)
Shortpage Properties

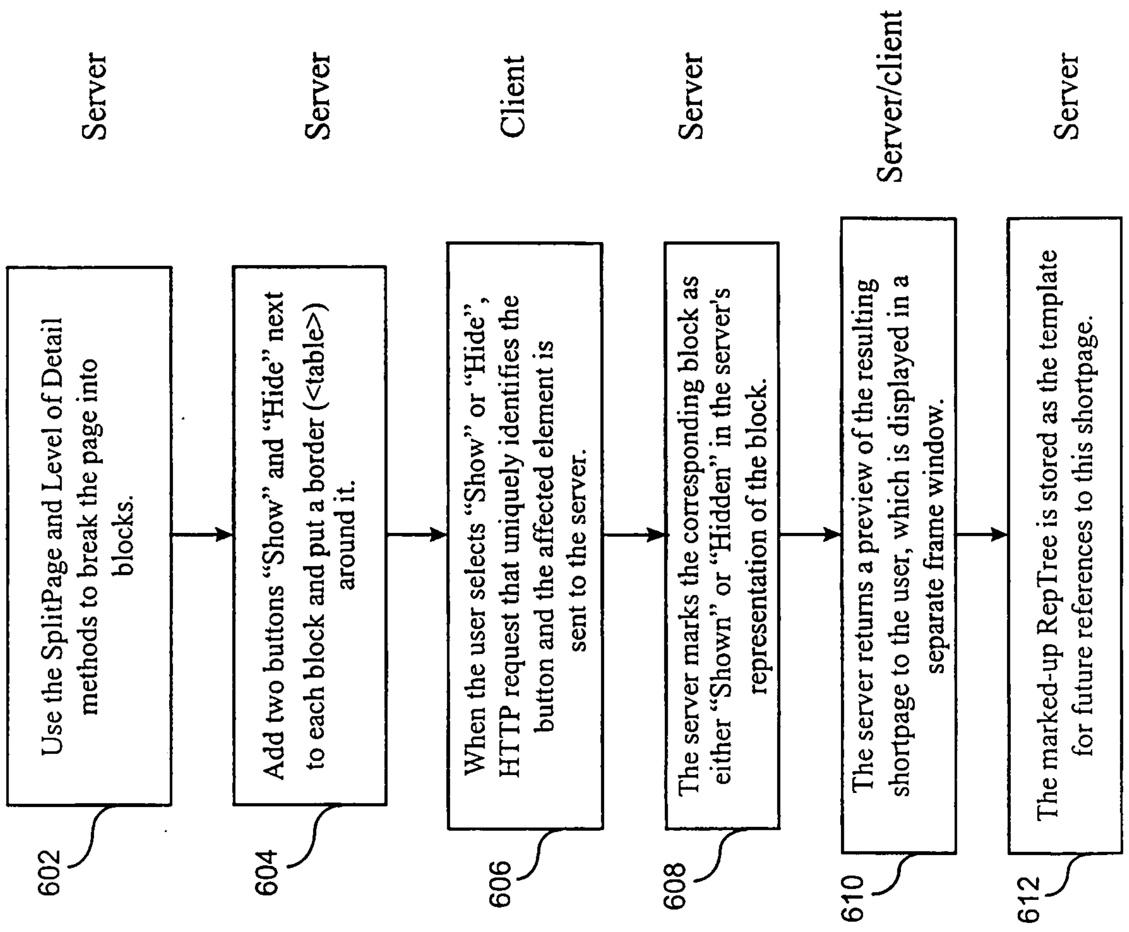
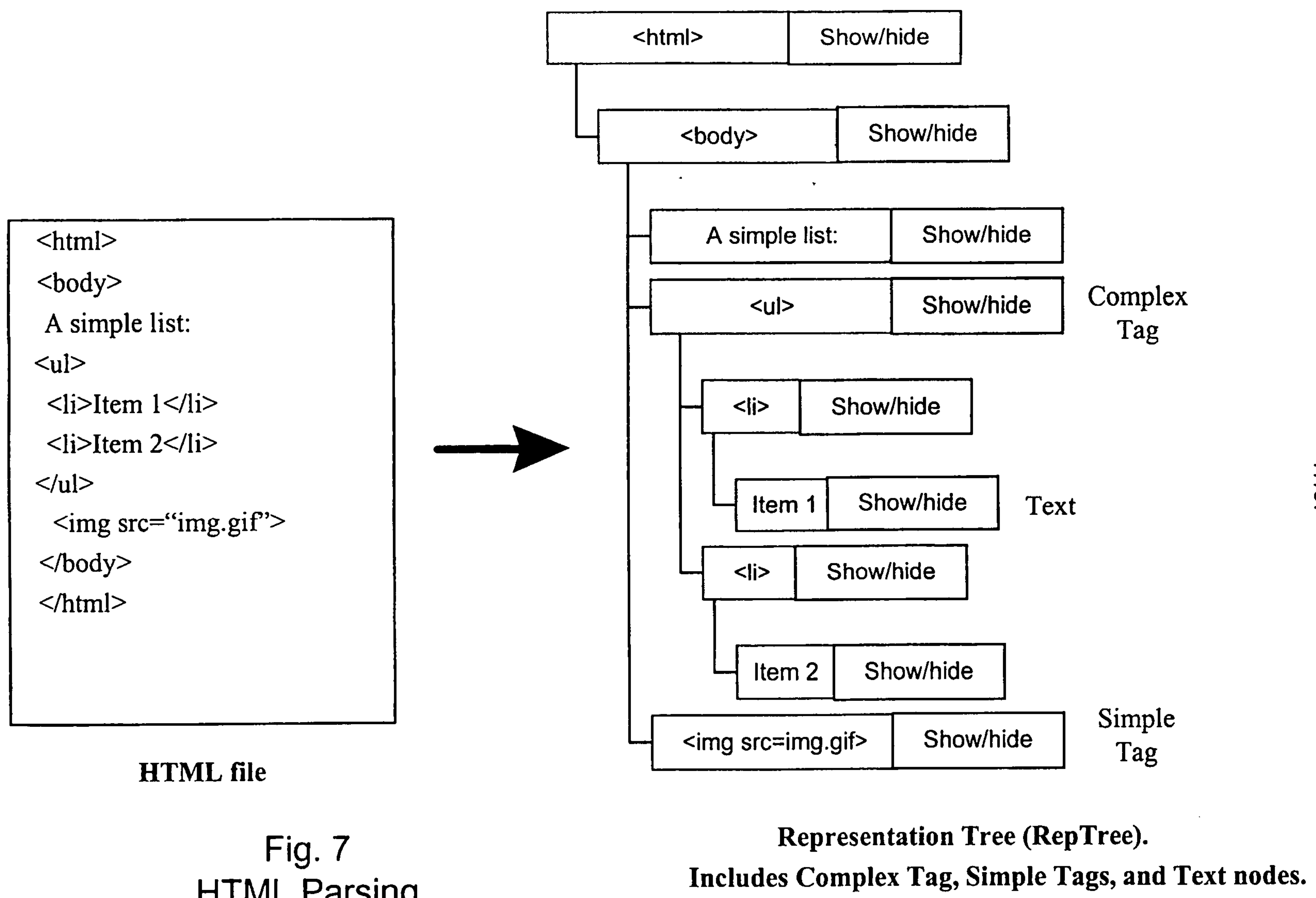

Server
Server
Client
Server
Server/client
Server
602
Use the SplitPage and Level of Detail methods to break the page into blocks.
604
Add two buttons "Show" and "Hide" next to each block and put a border (<table>) around it.
606
When the user selects "Show" or "Hide", HTTP request that uniquely identifies the button and the affected element is sent to the server.
608
The server marks the corresponding block as either "Shown" or "Hidden" in the server's representation of the block.
610
The server returns a preview of the resulting shortpage to the user, which is displayed in a separate frame window.
612
The marked-up RepTree is stored as the template for future references to this shortpage.

Fig. 6 Creating/Editing a Shortpage



```
<html>
<body>
 A simple list:
<ul>
 <li>Item 1</li>
 <li>Item 2</li>
</ul>
  <img src="img.gif">
</body>
</html>
```

Fig. 7
HTML Parsing

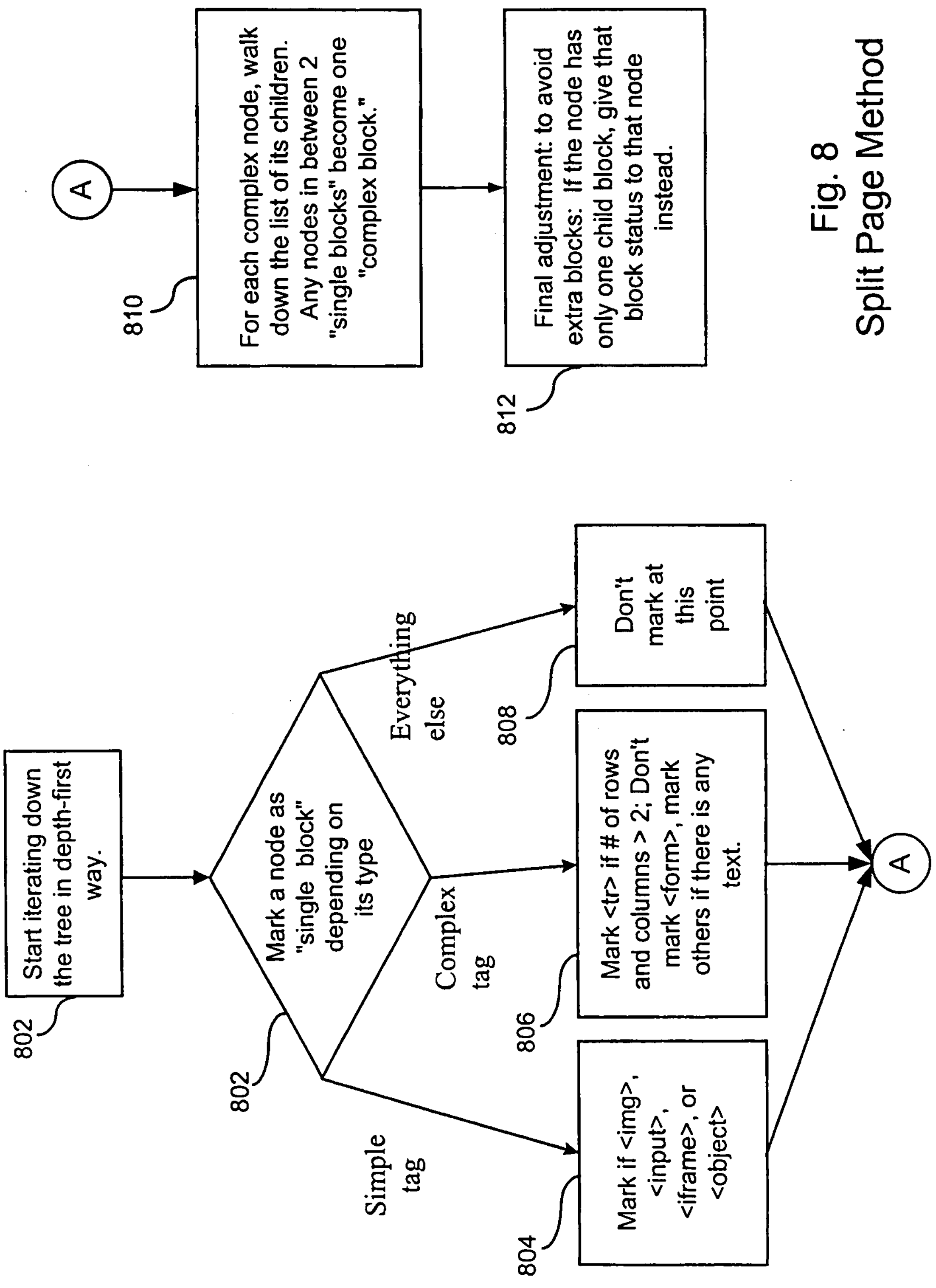
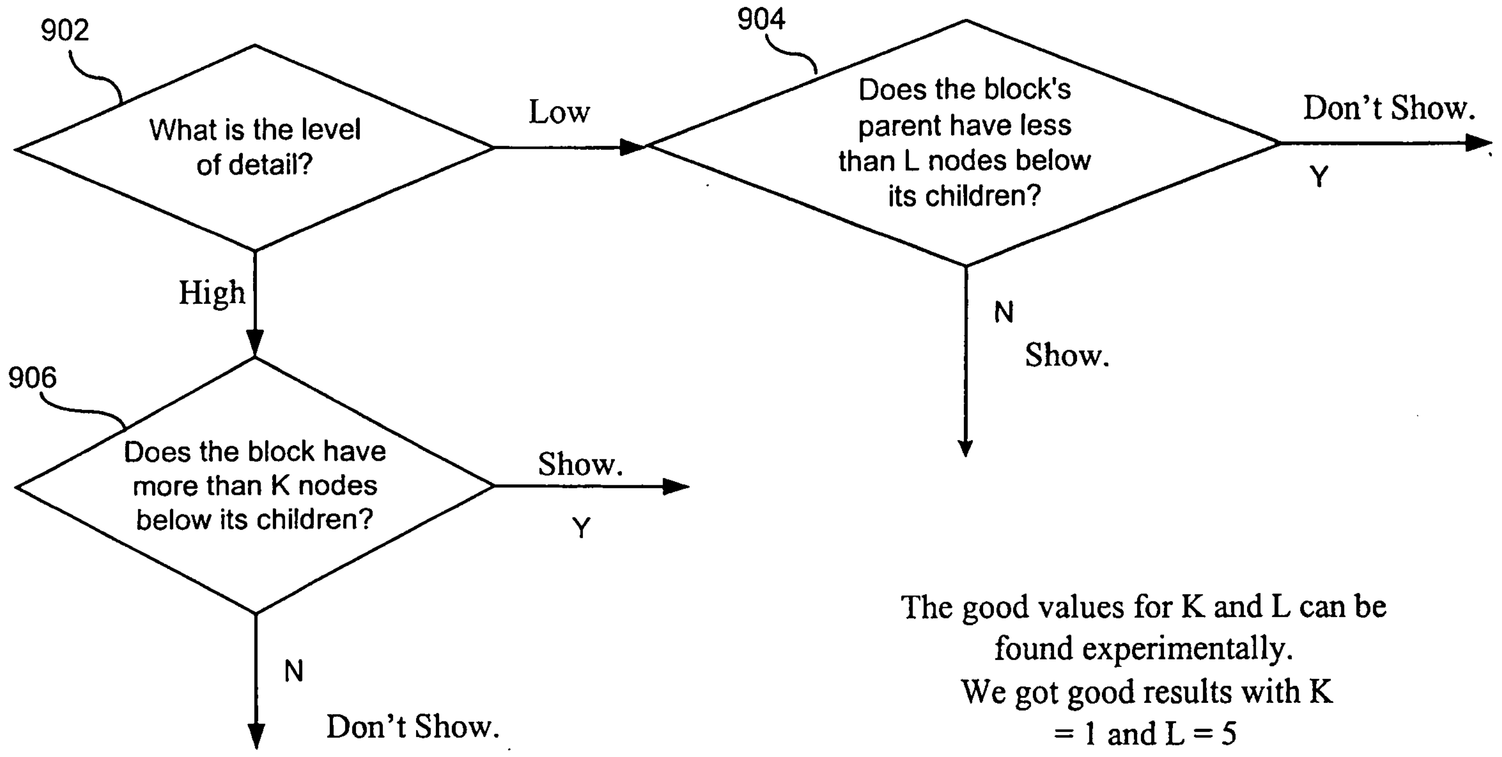

802
Start iterating down the tree in depth-first way.
802
Mark a node as "single block" depending on its type
Simple tag
Complex tag
Everything else
804
Mark if <img>, <input>, <iframe>, or <object>
806
Mark <tr> if # of rows and columns > 2; Don't mark <form>, mark others if there is any text.
808
Don't mark at this point
A
A
810
For each complex node, walk down the list of its children. Any nodes in between 2 "single blocks" become one "complex block."
812
Final adjustment: to avoid extra blocks: If the node has only one child block, give that block status to that node instead.

Fig. 8
Split Page Method


902
What is the level of detail?
Low
High
904
Does the block's parent have less than L nodes below its children?
Don't Show.
Y
N
Show.
906
Does the block have more than K nodes below its children?
Show.
Y
N
Don't Show.

The good values for K and L can be found experimentally.
We got good results with K = 1 and L = 5

Fig. 9

The level of detail feature is used to decrease the number of blocks displayed at any given time. Any number of levels can be supported. This flow chart demonstrates how to work with two levels.

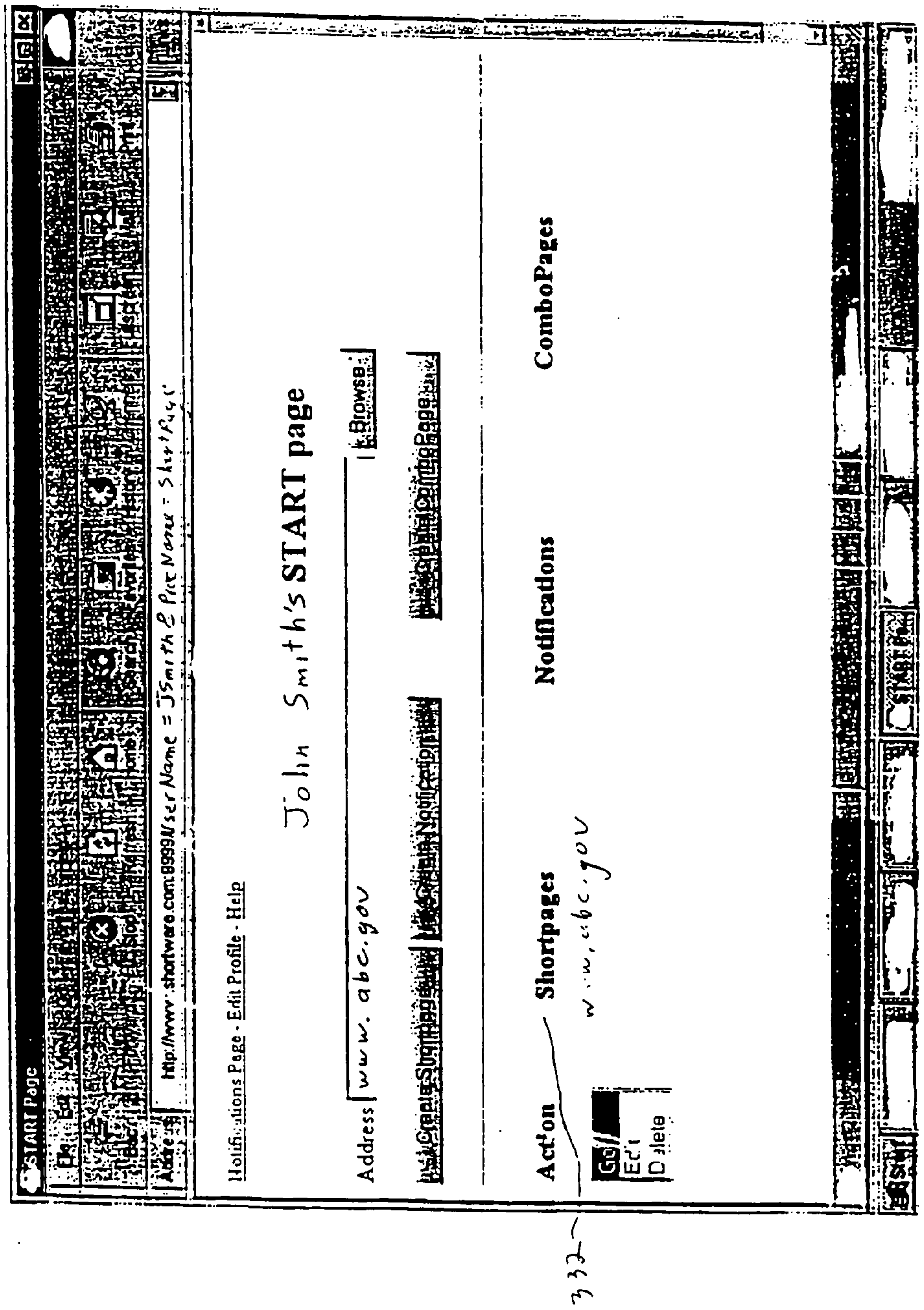


Fig. 10
After Creating a Shortpage

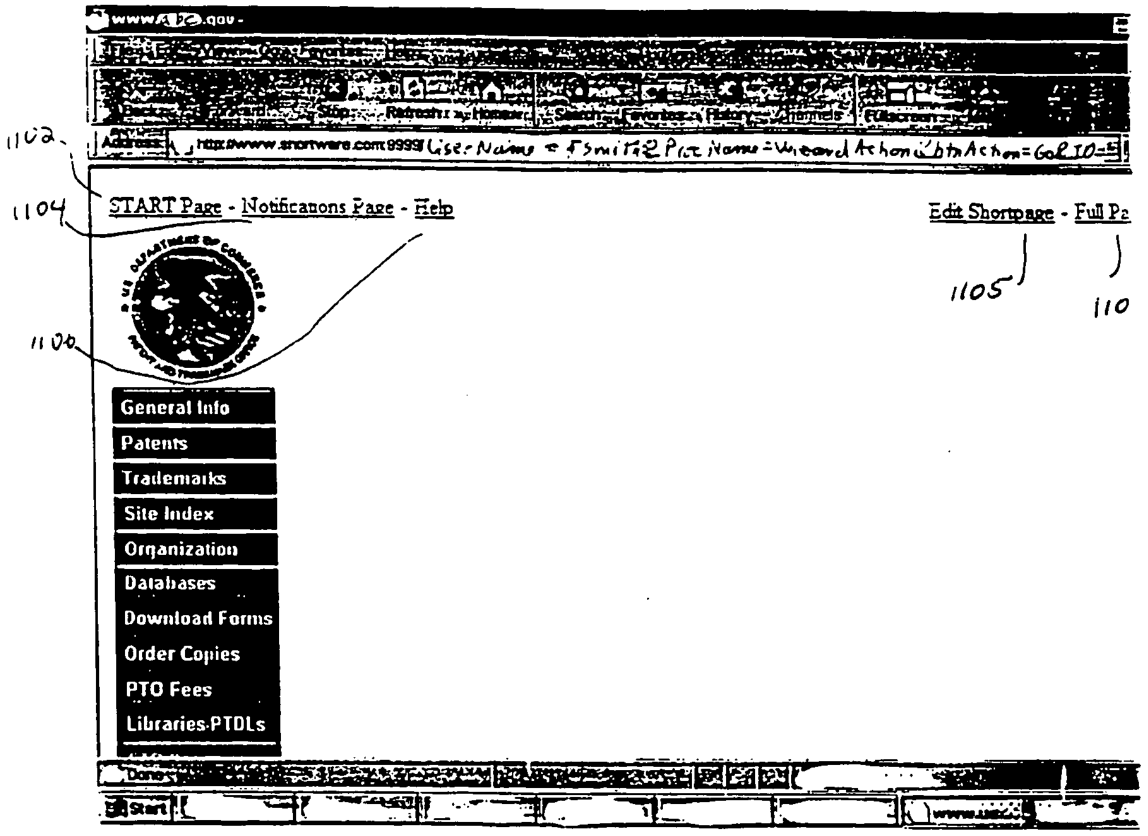
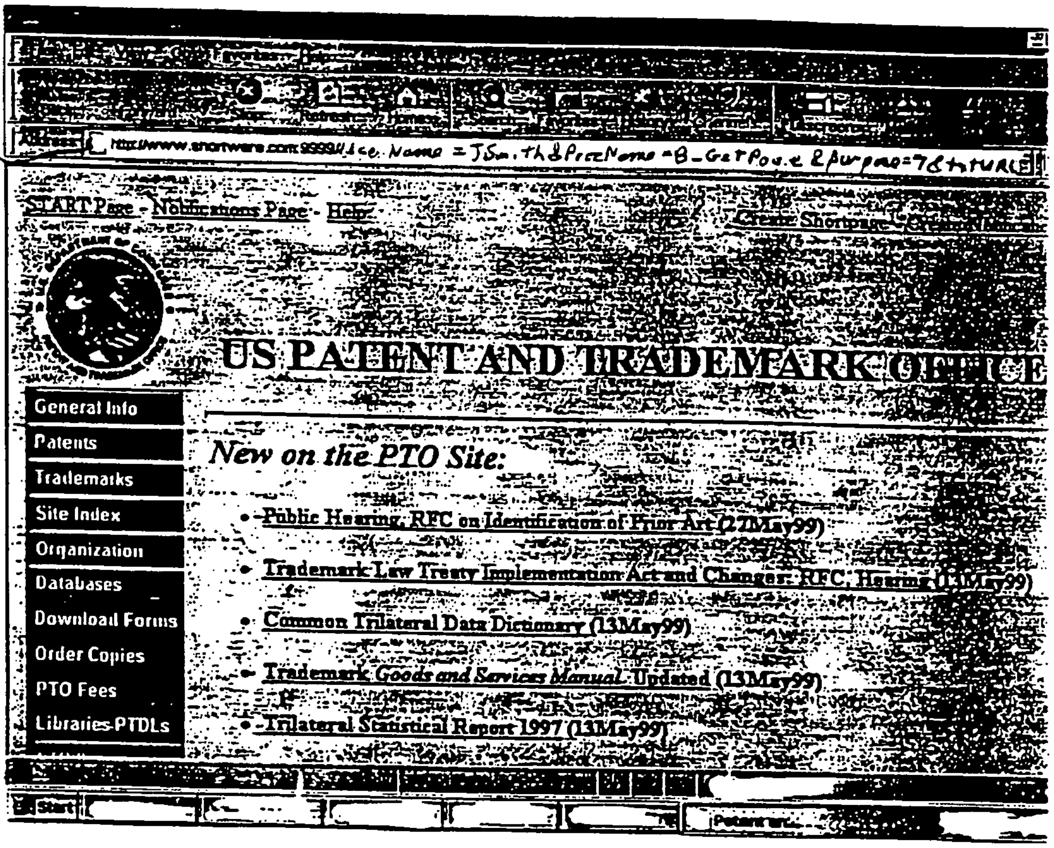


Fig. 11
Viewing a Shortpage

1202



Address http://www.shortware.com:9999/Use.Name=JSmith&PrezName=B_GetPage&purpose=7&httURL

START Page - Notifications Page - Help

Create Shortpage

US PATENT AND TRADEMARK OFFICE

General Info
Patents
Trademarks
Site Index
Organization
Databases
Download Forms
Order Copies
PTO Fees
Libraries-PTDLs

New on the PTO Site:

- Public Hearing, RFC on Identification of Prior Art (27May99)
- Trademark Law Treaty Implementation Act and Changes: RFC, Hearing (13May99)
- Common Trilateral Data Dictionary (13May99)
- Trademark *Goods and Services Manual* Updated (13May99)
- Trilateral Statistical Report 1997 (13May99)

Start

Fig. 12
Viewing a Fullpage

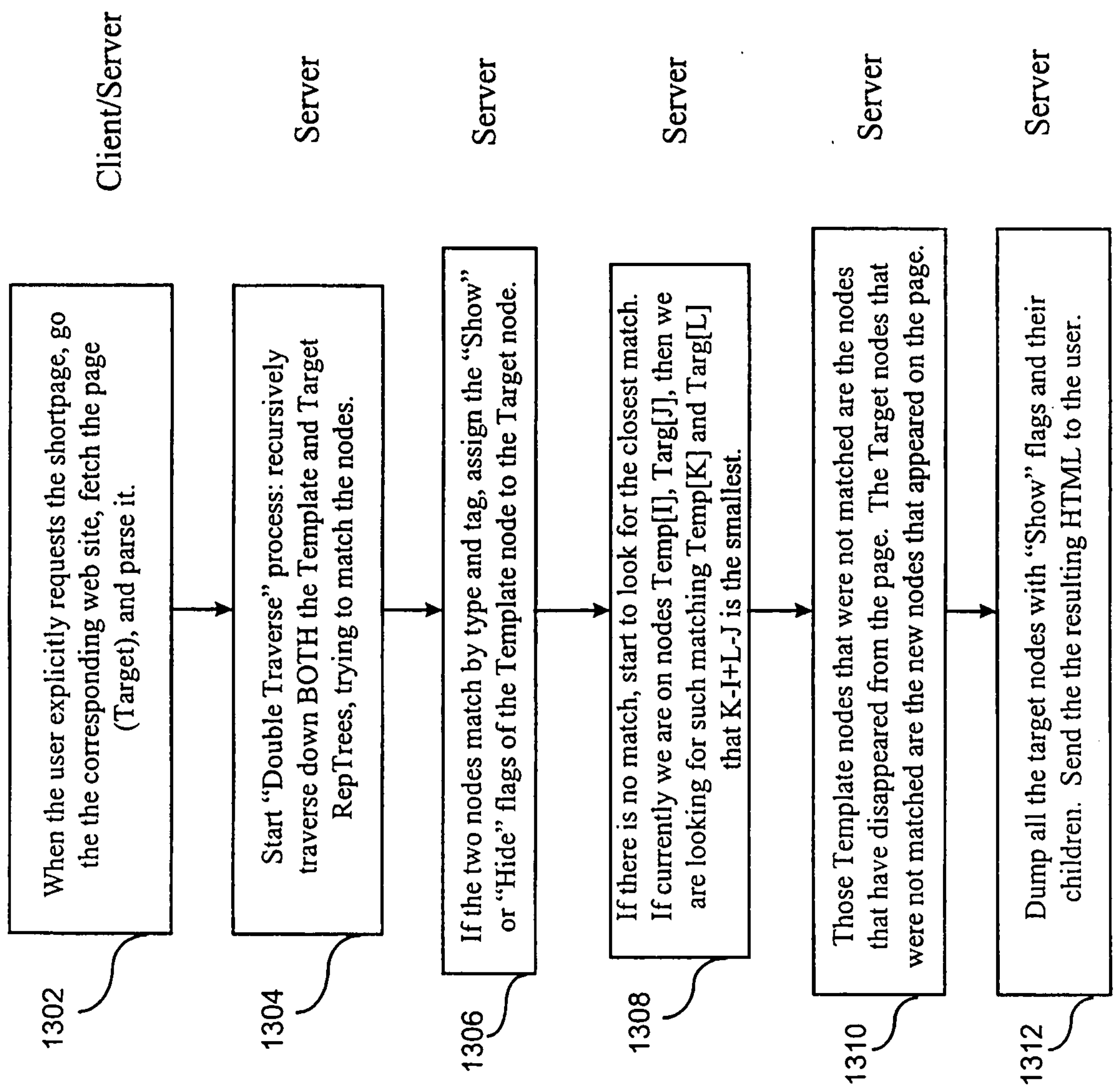

Client/Server
1302
When the user explicitly requests the shortpage, go the the corresponding web site, fetch the page (Target), and parse it.
Server
1304
Start "Double Traverse" process: recursively traverse down BOTH the Template and Target RepTrees, trying to match the nodes.
Server
1306
If the two nodes match by type and tag, assign the "Show" or "Hide" flags of the Template node to the Target node.
Server
1308
If there is no match, start to look for the closest match. If currently we are on nodes Temp[I], Targ[J], then we are looking for such matching Temp[K] and Targ[L] that K-I+L-J is the smallest.
Server
1310
Those Template nodes that were not matched are the nodes that have disappeared from the page. The Target nodes that were not matched are the new nodes that appeared on the page.
Server
1312
Dump all the target nodes with "Show" flags and their children. Send the the resulting HTML to the user.

Fig. 13
Viewing a Shortpage

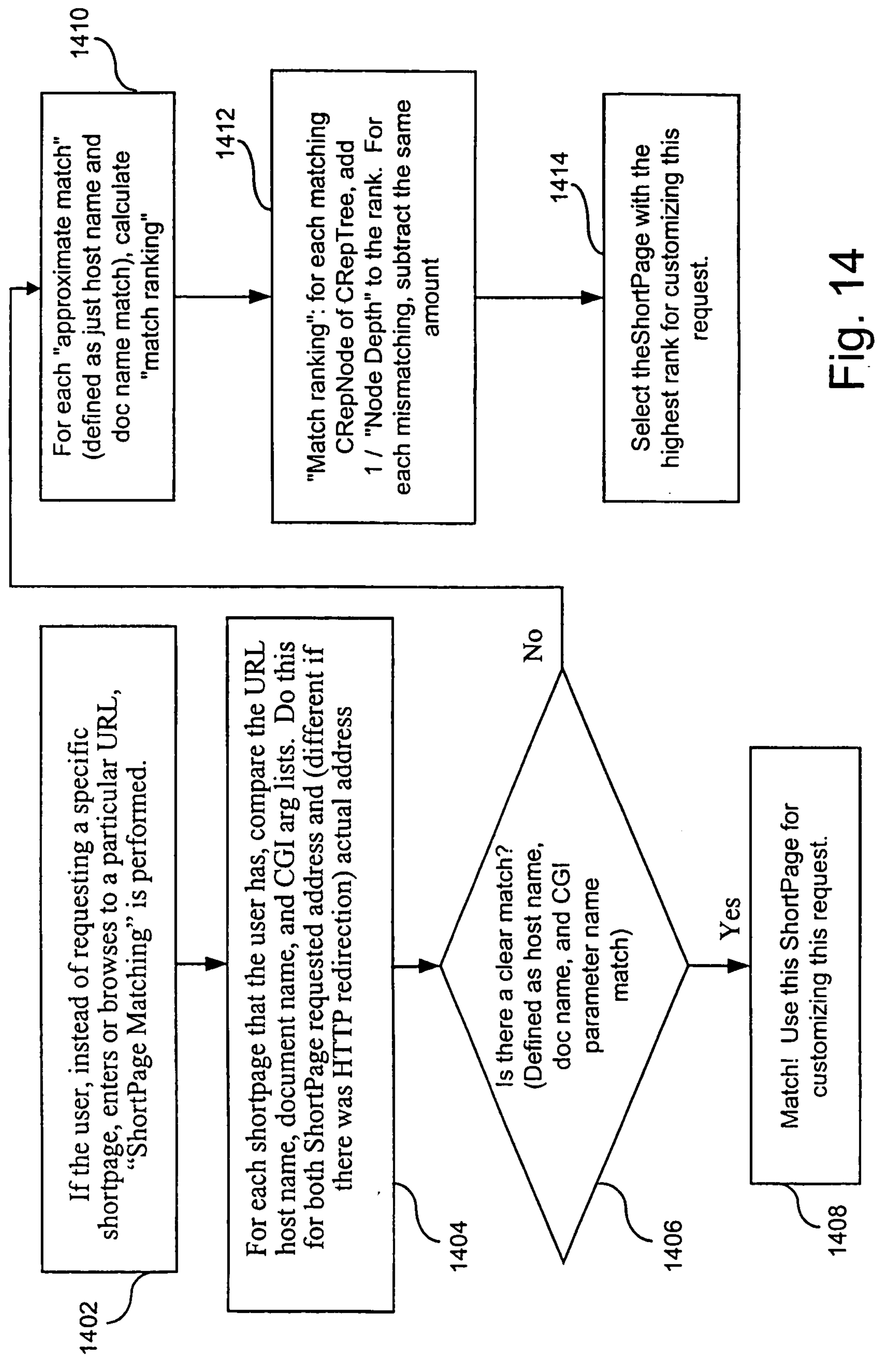

If the user, instead of requesting a specific shortpage, enters or browses to a particular URL, "ShortPage Matching" is performed.
1402
For each shortpage that the user has, compare the URL host name, document name, and CGI arg lists. Do this for both ShortPage requested address and (different if there was HTTP redirection) actual address
1404
Is there a clear match? (Defined as host name, doc name, and CGI parameter name match)
1406
Yes
No
Match! Use this ShortPage for customizing this request.
1408
For each "approximate match" (defined as just host name and doc name match), calculate "match ranking"
1410
"Match ranking": for each matching CRepNode of CRepTree, add 1 / "Node Depth" to the rank. For each mismatching, subtract the same amount
1412
Select theShortPage with the highest rank for customizing this request.
1414

Fig. 14
Shortpage Matching

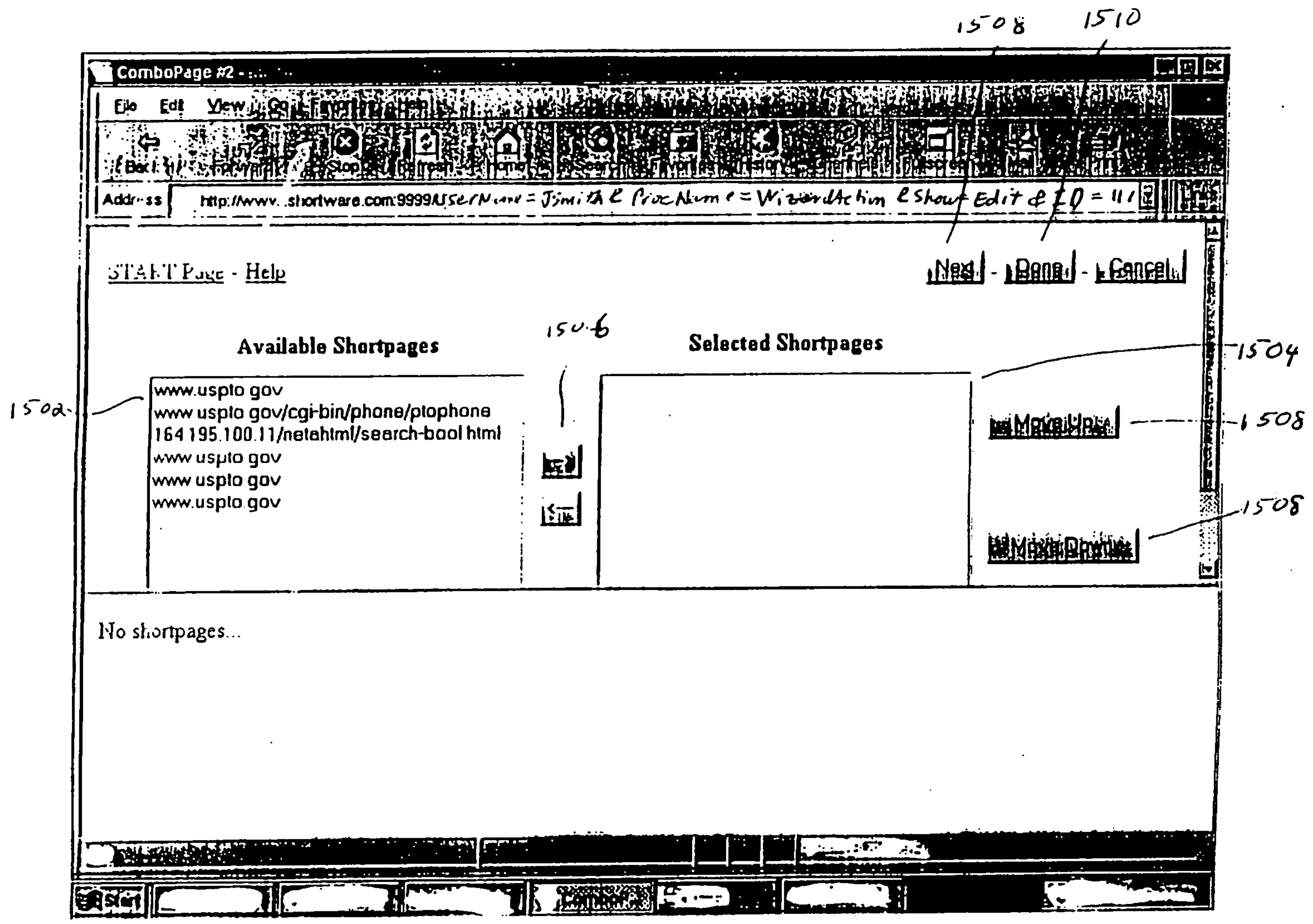
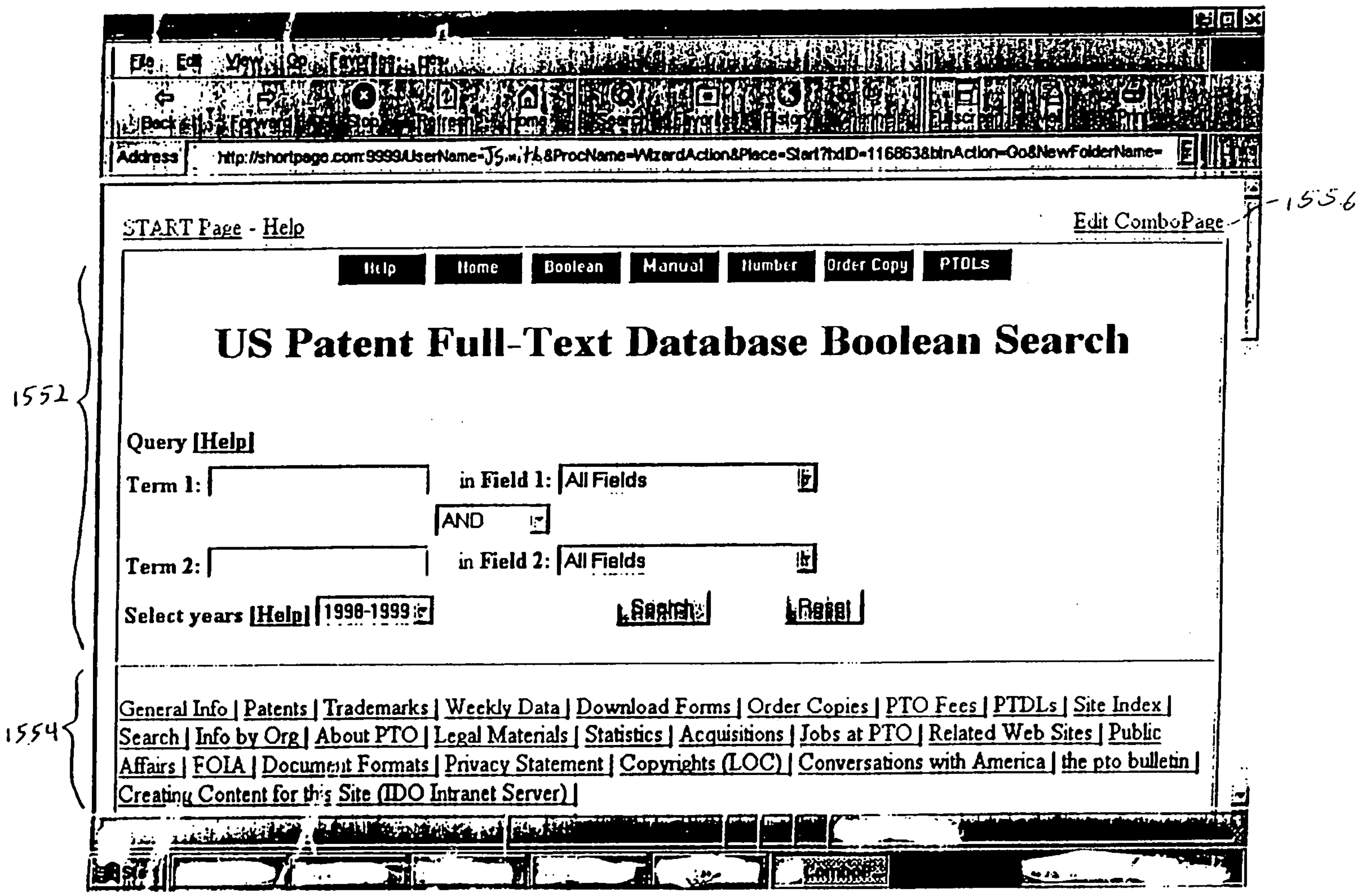

ComboPage #2
File Edit View
Back Forward Stop Refresh Home Search Favorites History
Address http://www..shortware.com:9999/UserName=Jsmith&ProcName=~~Wizard~~Action&Show=Edit&ID=111
START Page - Help
Next - Done - Cancel
Available Shortpages
www.uspto.gov
www.uspto.gov/cgi-bin/phone/ptophone
164.195.100.11/netahtml/search-bool.html
www.uspto.gov
www.uspto.gov
www.uspto.gov
Selected Shortpages
Move Up
Move Down
No shortpages...
Start
1502
1504
1506
1508
1508
1508
1510

Fig. 15(a)


File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History
Address
1552
1554
1556
Help Home Boolean Manual Number Order Copy PTOLs
Search Reset

http://shortpage.com:9999/UserName=JS...&ProcName=WizardAction&Place=Start?txID=116863&btnAction=Go&NewFolderName=

START Page - Help

Edit ComboPage

US Patent Full-Text Database Boolean Search

Query [Help]

Term 1: in Field 1: All Fields

AND

Term 2: in Field 2: All Fields

Select years [Help] 1998-1999

General Info | Patents | Trademarks | Weekly Data | Download Forms | Order Copies | PTO Fees | PTDLs | Site Index | Search | Info by Org | About PTO | Legal Materials | Statistics | Acquisitions | Jobs at PTO | Related Web Sites | Public Affairs | FOIA | Document Formats | Privacy Statement | Copyrights (LOC) | Conversations with America | the pto bulletin | Creating Content for this Site (IDO Intranet Server) |

Fig 15(b)

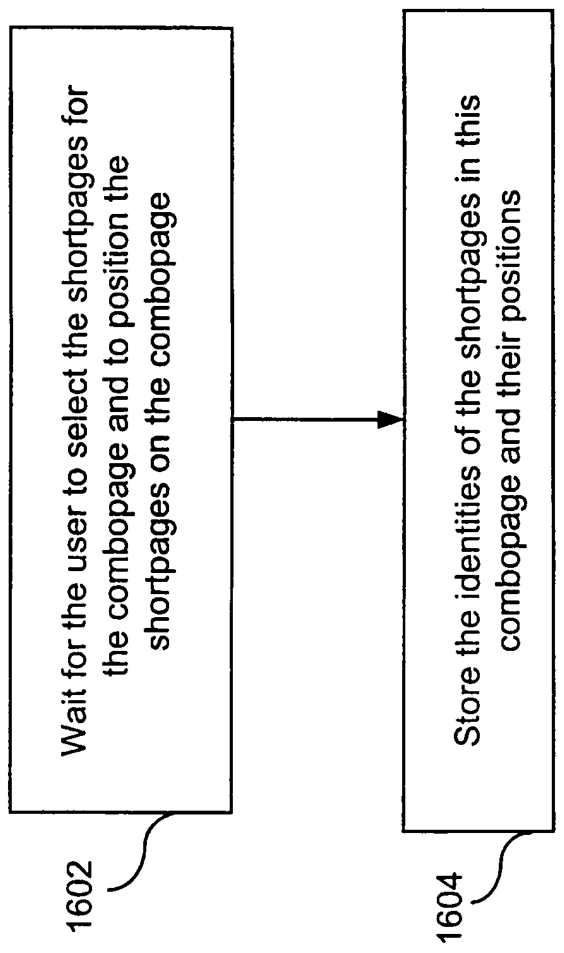

Wait for the user to select the shortpages for the combopage and to position the shortpages on the combopage
1602
Store the identities of the shortpages in this combopage and their positions
1604

Fig. 16(a)
Creating a Combopage

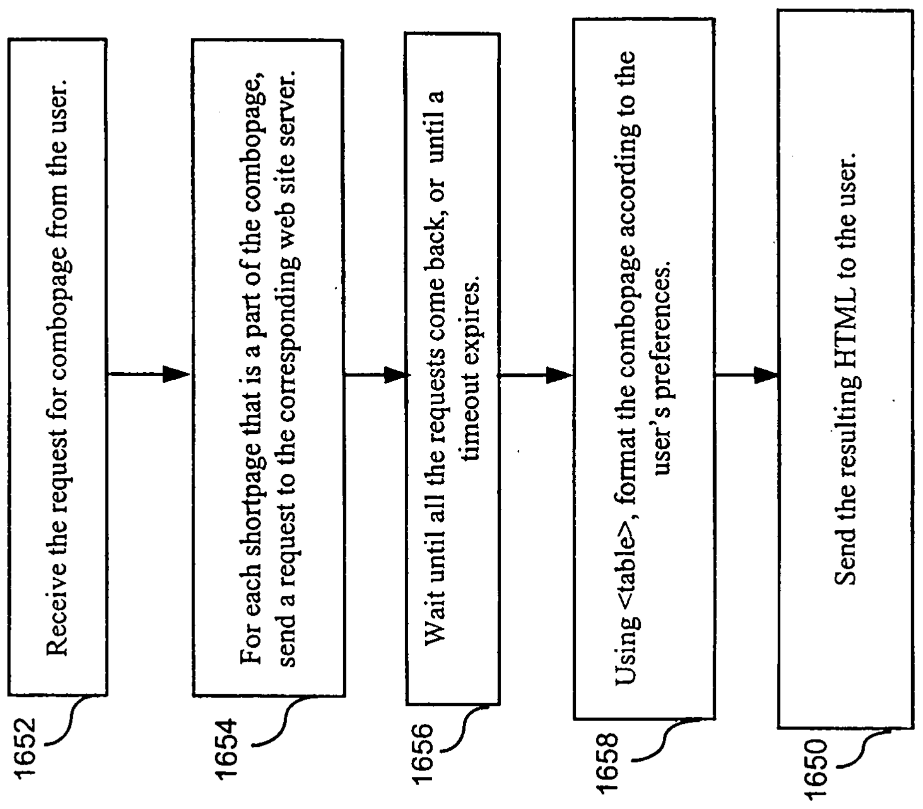
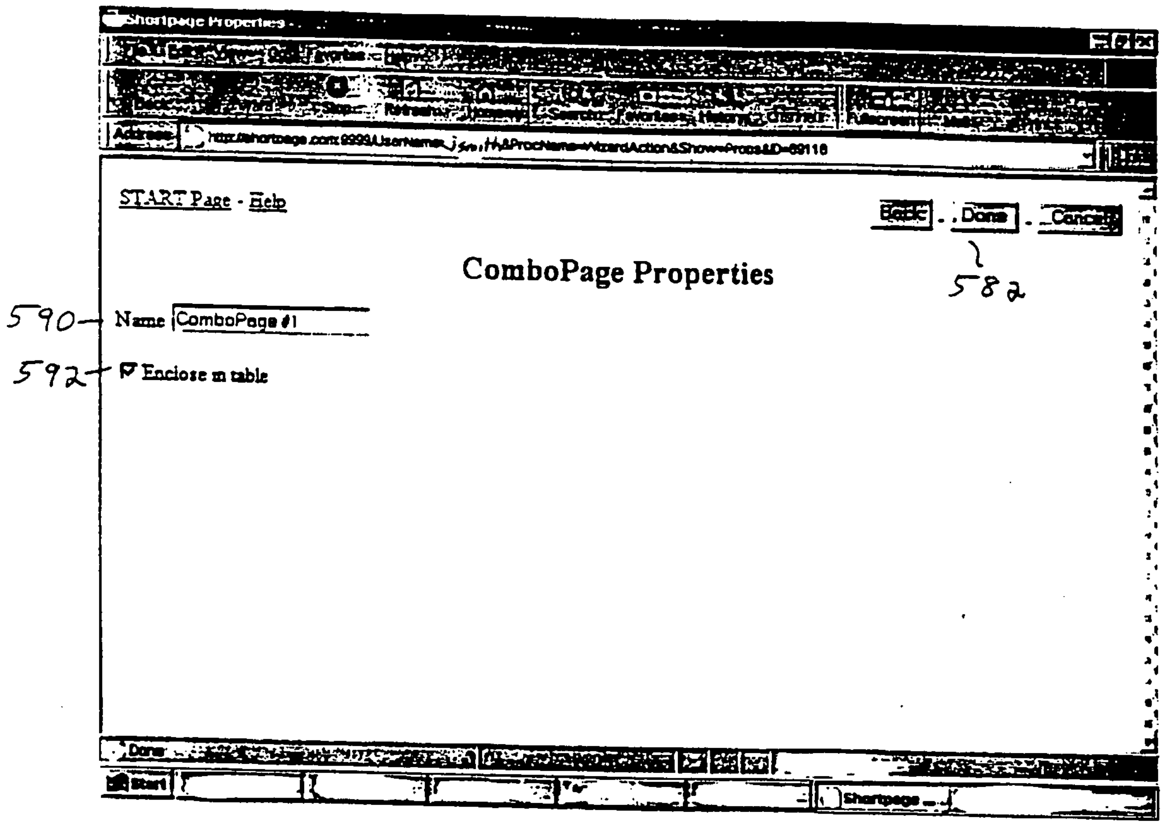

Receive the request for combopage from the user.
1652
For each shortpage that is a part of the combopage, send a request to the corresponding web site server.
1654
Wait until all the requests come back, or until a timeout expires.
1656
Using <table>, format the combopage according to the user's preferences.
1658
Send the resulting HTML to the user.
1650

Fig. 16(b) Displaying a Combopage (server side)


Shortpage Properties
Address http://shortpage.com:9999/UserName=jsmith&ProcName=WizardAction&Show=Props&ID=69118
START Page - Help
Back - Done - Cancel
582
590
592
Done
Start
Shortpage

ComboPage Properties

Name ComboPage #1

☑ Enclose in table

Fig. 16 (c)
Combopage Properties

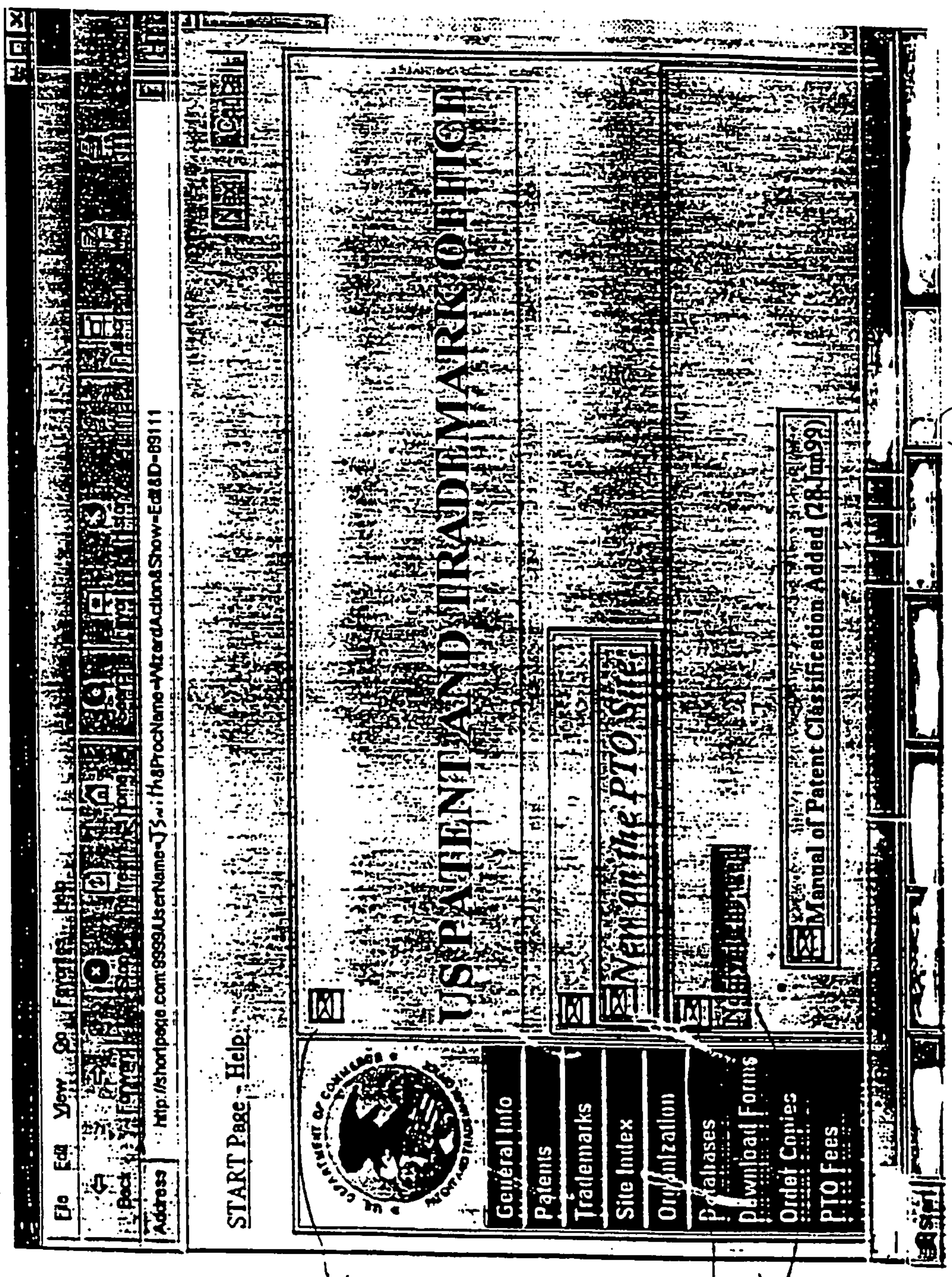
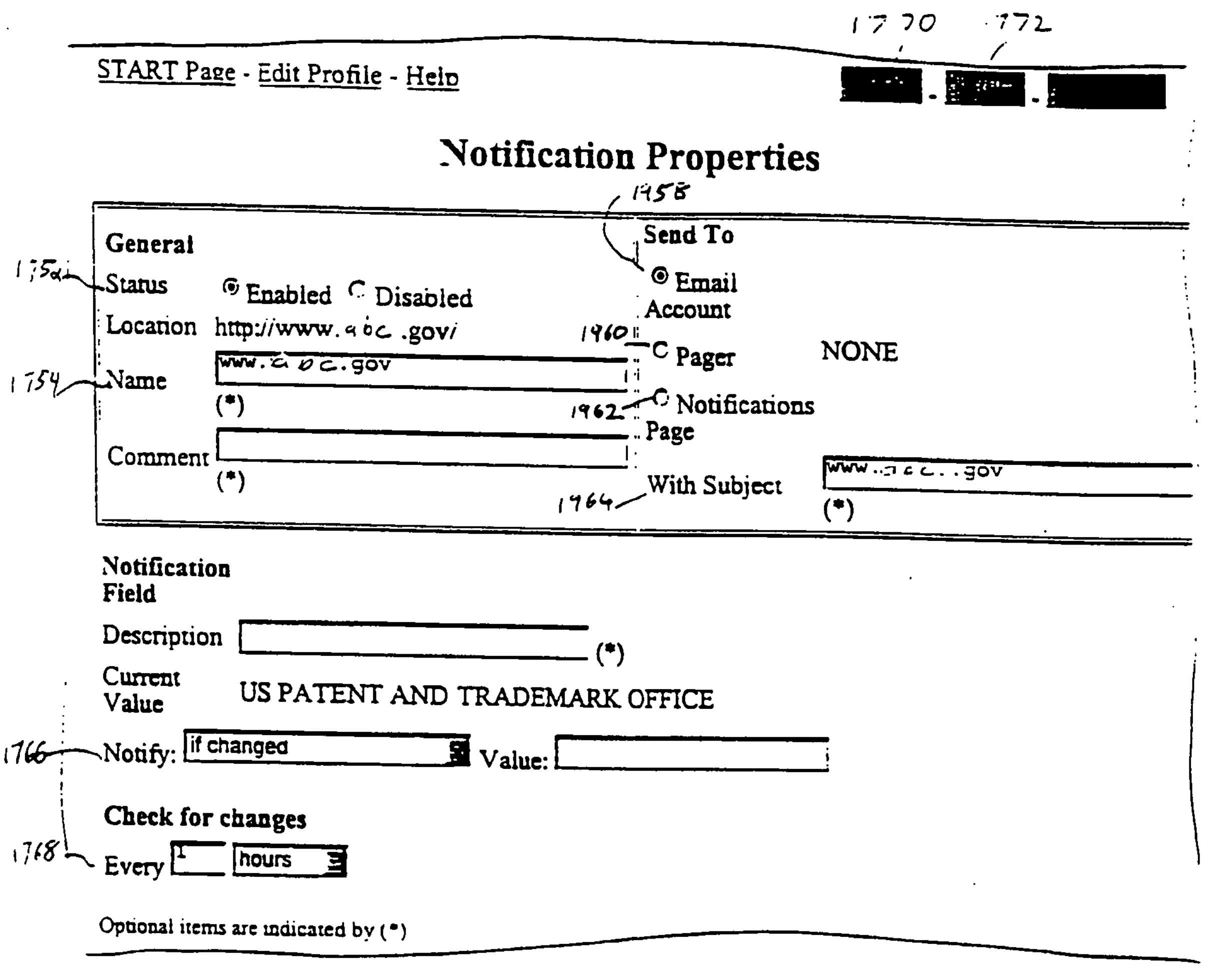

File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print
Address http://shortpage.com:8999/UserName=JS...&ProcName=WizardAction&Show=Edit&ID=89111
START Page - Help
US PATENT AND TRADEMARK OFFICE
General Info
Patents
Trademarks
Site Index
Organization
Databases
Download Forms
Order Copies
PTO Fees
News on the PTO Site
Manual of Patent Classification Added (28Jun99)
Start
1702
1704
1705

FIG. 17(a)



START Page - Edit Profile - Help

Notification Properties

General

Status [✓ Enabled] Disabled

Location http://www.abc.gov/

Name www.abc.gov

(*)

Comment

(*)

Send To

[✓ Email Account]

Pager NONE

Notifications Page

With Subject www.abc.gov

(*)

Notification Field

Description (*)

Current Value US PATENT AND TRADEMARK OFFICE

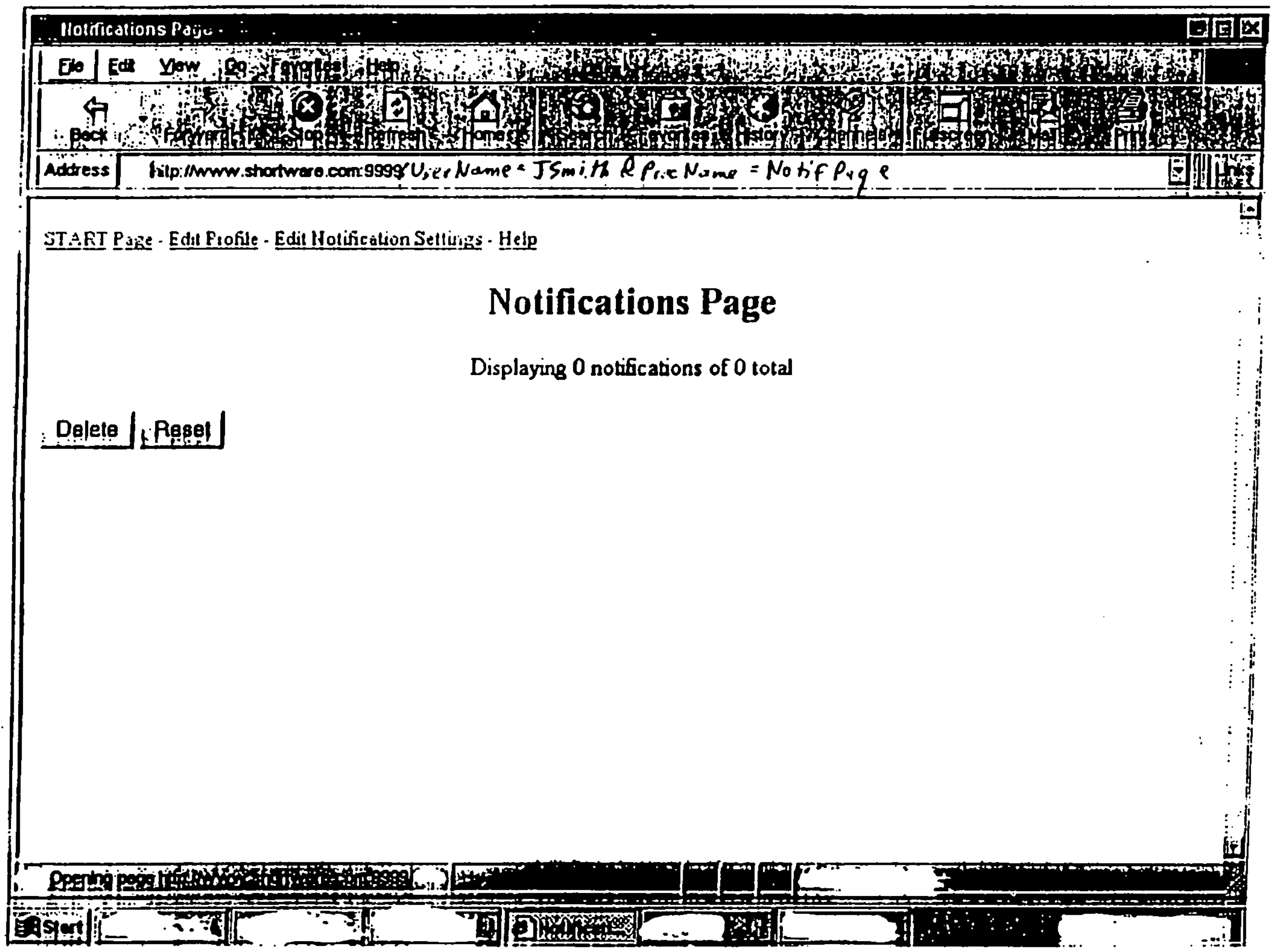
Notify: if changed Value:

Check for changes

Every 1 hours

Optional items are indicated by (*)

Fig. 17(b)


Notifications Page
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print
Address
Links
Delete
Reset
Start

http://www.shortware.com:9999/UserName=JSmith&PageName=NotifPage

START Page - Edit Profile - Edit Notification Settings - Help

Notifications Page

Displaying 0 notifications of 0 total

FIG. 17(C)

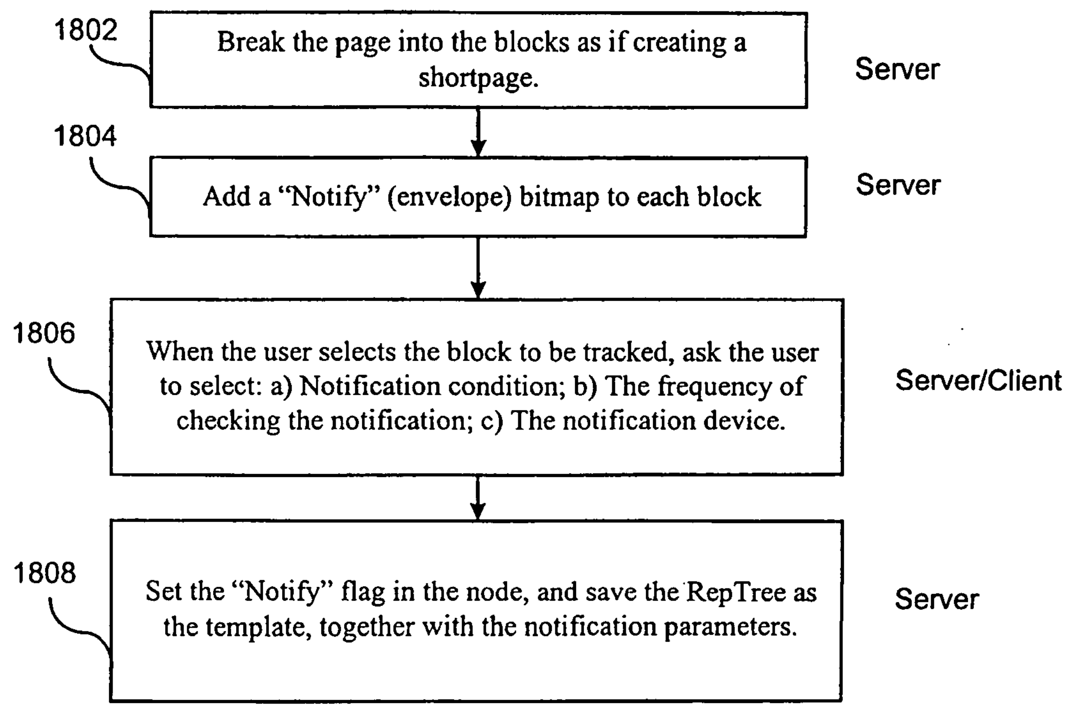

1802
Break the page into the blocks as if creating a shortpage.
Server
1804
Add a "Notify" (envelope) bitmap to each block
Server
1806
When the user selects the block to be tracked, ask the user to select: a) Notification condition; b) The frequency of checking the notification; c) The notification device.
Server/Client
1808
Set the "Notify" flag in the node, and save the RepTree as the template, together with the notification parameters.
Server

Fig. 18(a)
Setting Up
Notification

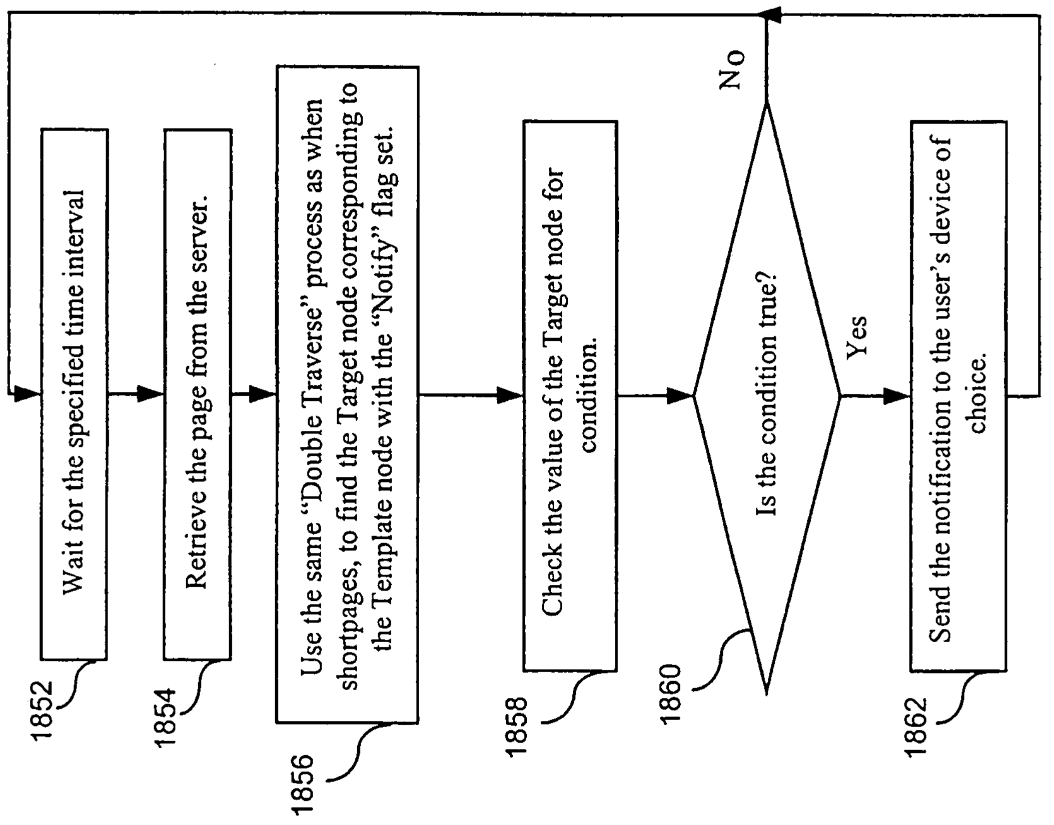

1852
Wait for the specified time interval
1854
Retrieve the page from the server.
1856
Use the same "Double Traverse" process as when shortpages, to find the Target node corresponding to the Template node with the "Notify" flag set.
1858
Check the value of the Target node for condition.
1860
Is the condition true?
No
Yes
1862
Send the notification to the user's device of choice.

Fig. 18(b) Issuing a Notification (server side)

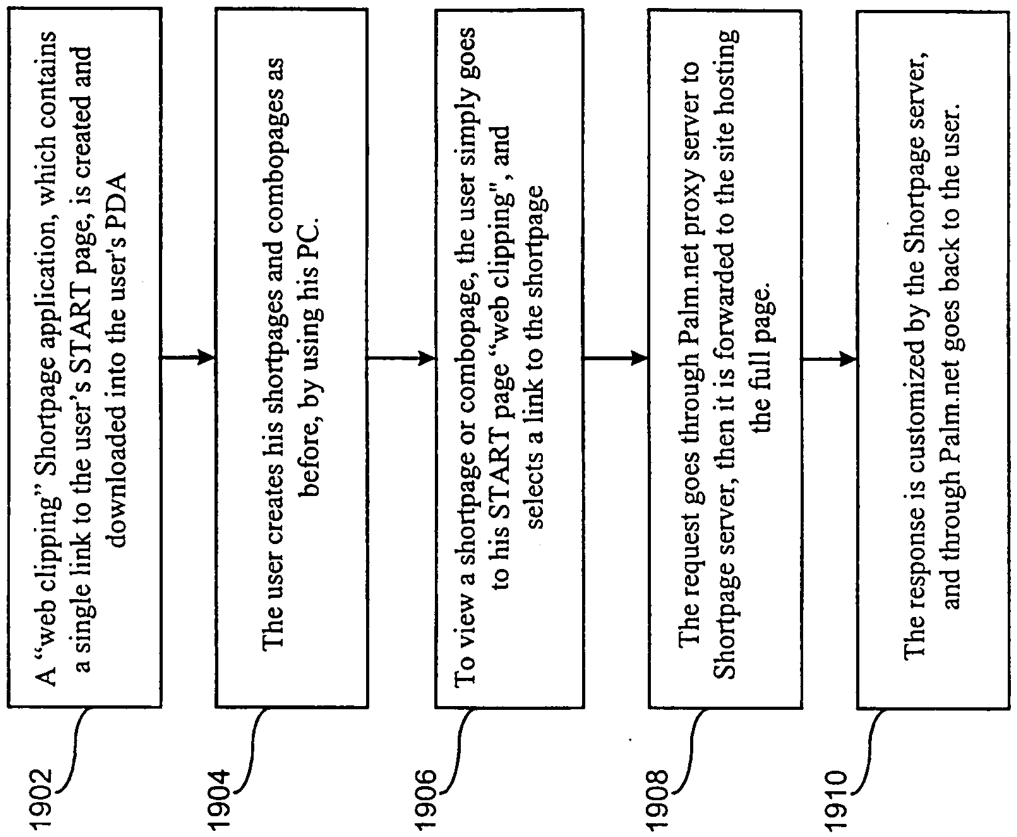

A "web clipping" Shortpage application, which contains a single link to the user's START page, is created and downloaded into the user's PDA
1902
The user creates his shortpages and combopages as before, by using his PC.
1904
To view a shortpage or combopage, the user simply goes to his START page "web clipping", and selects a link to the shortpage
1906
The request goes through Palm.net proxy server to Shortpage server, then it is forwarded to the site hosting the full page.
1908
The response is customized by the Shortpage server, and through Palm.net goes back to the user.
1910

Fig. 19
Mobile Client

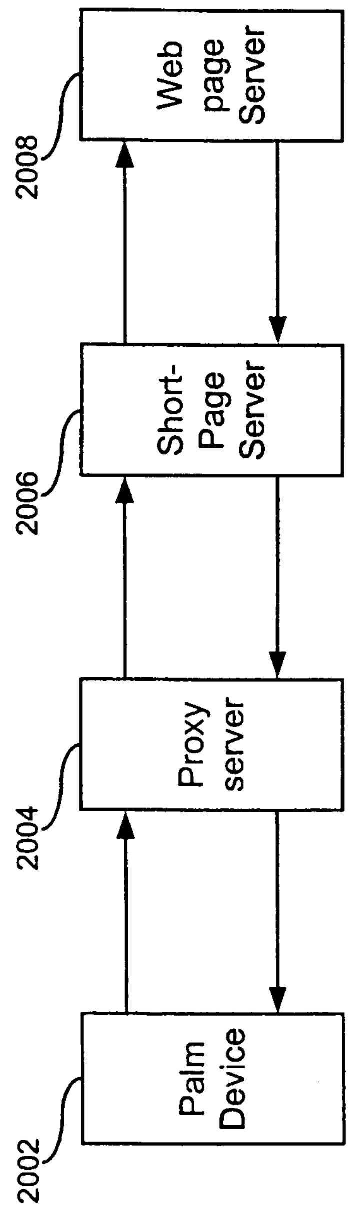
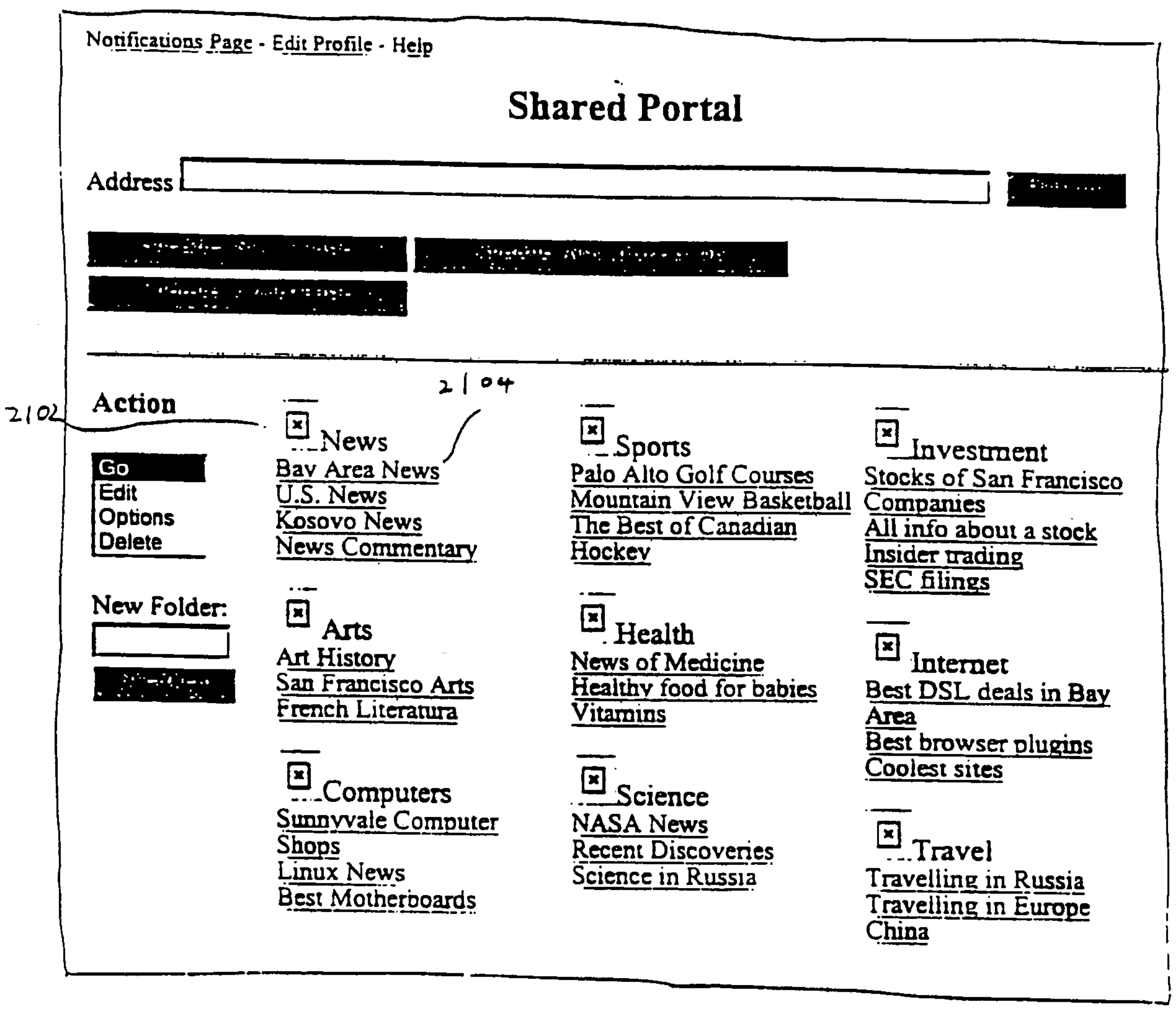

2002
Palm Device
2004
Proxy server
2006
Short-Page Server
2008
Web page Server

Fig. 20



Notifications Page - Edit Profile - Help

Shared Portal

Address

Action

Go
Edit
Options
Delete

New Folder:

News
Bay Area News
U.S. News
Kosovo News
News Commentary

Arts
Art History
San Francisco Arts
French Literatura

Computers
Sunnyvale Computer Shops
Linux News
Best Motherboards

Sports
Palo Alto Golf Courses
Mountain View Basketball
The Best of Canadian Hockey

Health
News of Medicine
Healthy food for babies
Vitamins

Science
NASA News
Recent Discoveries
Science in Russia

Investment
Stocks of San Francisco Companies
All info about a stock
Insider trading
SEC filings

Internet
Best DSL deals in Bay Area
Best browser plugins
Coolest sites

Travel
Travelling in Russia
Travelling in Europe
China

Fig. 21